International Telecommunication Union

# ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

# G.9954

(02/2005)

SERIES G: TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

Access networks – In premises networks

## Phoneline networking transceivers – Enhanced physical, media access, and link layer specifications

ITU-T Recommendation G.9954

ITU-T G-SERIES RECOMMENDATIONS

**TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS**

| | |
|---|---|
| INTERNATIONAL TELEPHONE CONNECTIONS AND CIRCUITS | G.100–G.199 |
| GENERAL CHARACTERISTICS COMMON TO ALL ANALOGUE CARRIER-TRANSMISSION SYSTEMS | G.200–G.299 |
| INDIVIDUAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON METALLIC LINES | G.300–G.399 |
| GENERAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON RADIO-RELAY OR SATELLITE LINKS AND INTERCONNECTION WITH METALLIC LINES | G.400–G.449 |
| COORDINATION OF RADIOTELEPHONY AND LINE TELEPHONY | G.450–G.499 |
| TRANSMISSION MEDIA CHARACTERISTICS | G.600–G.699 |
| DIGITAL TERMINAL EQUIPMENTS | G.700–G.799 |
| DIGITAL NETWORKS | G.800–G.899 |
| DIGITAL SECTIONS AND DIGITAL LINE SYSTEM | G.900–G.999 |
| QUALITY OF SERVICE AND PERFORMANCE – GENERIC AND USER-RELATED ASPECTS | G.1000–G.1999 |
| TRANSMISSION MEDIA CHARACTERISTICS | G.6000–G.6999 |
| DATA OVER TRANSPORT – GENERIC ASPECTS | G.7000–G.7999 |
| ETHERNET OVER TRANSPORT ASPECTS | G.8000–G.8999 |
| ACCESS NETWORKS | G.9000–G.9999 |
| **In premises networks** | **G.9950–G.9999** |

*For further details, please refer to the list of ITU-T Recommendations.*

# ITU-T Recommendation G.9954

## Phoneline networking transceivers – Enhanced physical, media access, and link layer specifications


**Summary**

This Recommendation defines the PHY, MAC, LINK and CONVERGENCE protocol stack layers for the G.9954 system providing the following features:

- PHY-layer payload transmission rates of 4 to 240 Mbit/s;
- Rate adaptive transceivers that optimize data rates and packet error rates for dynamically varying channel conditions on a per-packet basis;
- Frequency Diverse QAM for robust communication over highly frequency-selective channels;
- Spectrum notching for compatibility with Amateur Radio services;
- Synchronous MAC protocol controlled by a dynamically elected master employing a combination of collision avoidance and controlled contention-based access strategies;
- Support for isochronous and asynchronous data services;
- Peer-to-peer communication within a master-controlled network;
- Master-less mode of operation using G.9951/2-like asynchronous MAC protocol;
- Packet aggregation (packetization) performed within the G.9954 protocol stack layer up to latency limits of the service flow and available transmission bandwidth;
- Quality of Service guarantees for bandwidth, jitter, latency and BER;
- QoS support for services with explicit traffic and rate specifications providing a link layer that is well suited for streaming audio and video;
- Protocol-specific convergence layers;
- Backward Compatible with G.9951/2, allowing transmissions at G.9951/2 compatibility mode rates;
- Coexistence and interoperability between G.9951/2 and G.9954 devices in a mixed network;
- Compatibility with other phoneline services such as POTS, V.90, ISDN and G.992.1, G.992.2, G.992.3, and G.992.4;
- Local and remote management of G.9954 devices;
- Provisions for future security extensions.



**Source**

ITU-T Recommendation G.9954 was approved on 13 February 2005 by ITU-T Study Group 15 (2005-2008) under the ITU-T Recommendation A.8 procedure.

## FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

## NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g. interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

## INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

## CONTENTS

# ITU-T Recommendation G.9954

## Phoneline networking transceivers – Enhanced physical, media access, and link layer specifications

### 1 Scope

This Recommendation specifies the interoperability and compatibility for G.9954 stations. The requirements are written from the perspective of a compliant transmitter, although some minimum performance requirements are established for receivers. This Recommendation does not specify implementation.

The structure of this Recommendation is as follows:

- **Clause 6: PHY layer specification –** This clause specifies the G.9954 PHY layer specification.
- **Clause 7: Media Access Protocol Specification** – This clause specifies the G.9954 Media Access Protocol including Asynchronous and Synchronous MAC modes of operation.
- **Clause 8: Compatibility Specification** – This clause describes the method by which backwards compatibility, coexistence and interoperability with G.9951/2 nodes is achieved in a mixed network of G.9951/2 and G.9954 nodes.
- **Clause 9: Quality of Service** – This clause describes the G.9954 Quality of Service framework.
- **Clause 10: Link Layer Protocol Specification** – This clause specifies the required link layer control functionalities.

### 2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation

[1] ITU-T Recommendation G.9951 (2001) (Formerly G.989.1), *Phoneline networking transceivers – Foundation*.

[2] ITU-T Recommendation G.9952 (2001) (Formerly G.989.2), *Phoneline networking transceivers – Payload format and link layer requirements*.

[3] ITU-T Recommendation G.9953 (2003) (Formerly G.989.3), *Phoneline networking transceivers – Isolation function*.

### 3 Terms and definitions

This Recommendation defines the following terms:

**3.1 AMAC mode**: The media access method used by a G.9954 device when operating in a network that does not contain a master.

**3.2 BACKOFF20**: The 20-symbol sequence used for signalling in the backoff slots, consisting of the TRN16 sequence followed by the EOF sequence.

**3.3 BACKOFF20 signal**: A symbol sequence that active transmitters can send in the three signal slots that follow a collision. Used in the distributed backoff level algorithm.

**3.4 binary exponential backoff (BEB)**: The IEEE Std 802.3 method for collision resolution.

**3.5 bounded DFPQ**: A collision resolution method, based on distributed fair priority queuing (DFPQ), where the collision resolution cycle is bounded within a transmission opportunity.

**3.6 broadcast packet**: A packet with the all-ones destination address (FF.FF.FF.FF.FF.FF).

**3.7 capability and status announcement**: A link-layer control protocol that is used to flood status information between stations with low overhead.

**3.8 collision fragment**: The fixed transmission sequence consisting of preamble, frame header, DA, SA, ET and EOF.

**3.9 contention period**: A media access period where devices contend for access to the media using priority-based CSMA/CD and collision resolution techniques.

**3.10 contention-free period**: A media access period, allocated to a single network device, in which media access contention and collisions should not (normally) occur.

**3.11 convergence layer**: A protocol-specific sub-layer that maps transport layer protocols into the native primitives of the G.9954 link layer.

**3.12 CS_IFG**: The minimum amount of media silence that must be guaranteed between consecutive frame bursts.

**3.13 device ID**: A unique identifier allocated to a G.9954 device by the master after registration.

**3.14 endpoint**: A G.9954 device that is not the master.

**3.15 EOF sequence**: The 4-symbol sequence that is appended to the physical layer frame, consisting of the first four symbols of the TRN sequence.

**3.16 flow**: A unidirectional flow of data between network nodes characterized by traffic with well-defined QoS parameters for throughput, latency, jitter and BER.

**3.17 flow ID**: A unique identifier of a flow between a source and destination device.

**3.18 flow signalling**: A G.9954 link layer protocol used to set up, modify and tear down flows.

**3.19 flow specification**: A specification of the characteristics of a flow in terms of its QoS traffic and rate parameters.

**3.20 G.995x**: A general reference to current PNT technology.

**3.21 G.9954**: A reference for the enhanced PNT technology proposed in this Recommendation.

**3.22 jitter**: A measure of the latency variation above and below a mean latency value. The maximum jitter is defined as the maximum latency variation above and below the mean latency value and is expressed as (+Max/–Min).

**3.23 latency**: A measure of the delay from the point in time when a packet reaches the service access point of the PNT protocol stack until the last bit of the packet has been transmitted successfully on the wire. Mean and maximum latency measurements are assumed to be calculated over the 99th percentile of all latency measurements.

**3.24 link integrity**: A background process that derives a user indication that the interface is attached to the phoneline and can detect at least one other station.

**3.25 link level priority**: The software priority class associated with the link-layer packet. This value may be mapped when converting to/from PHY Priority.

**3.26 MAC cycle**: The media access period between two consecutive transmissions of the MAP control frame.

**3.27 MAP**: A control frame describing the media access plan for the following MAC cycle.

**3.28 MAP_IFG**: The amount of media silence between frame bursts used by the master in the media access planning and advertised in the MAP control frame.

**3.29 master**: A G.9954 device that has master-capabilities and was selected as the current active master. The master is responsible for controlling the synchronous MAC mode of operation by planning media access timing on the network and periodically advertising the media access plan to all devices on the network.

**3.30 master-controlled network**: A network that contains a G.9954 device that is acting in the role of master. Media access by G.9954 devices in a master-controlled network is performed according to SMAC media access rules.

**3.31 master-less network**: A network that does not contain a device that is currently acting in the role of master. Media access by G.9954 devices in a master-less network is performed according to AMAC media access rules.

**3.32 packet aggregation**: The concatenation of transport and link layer packets into a single PHY frame burst.

**3.33 payload encoding**: The spectral mask, baud and the constellation encoding (bits-per-symbol) of the payload bits.

**3.34 PHY priority**: The 3-bit absolute priority used by the G.9951/2 media access control to rank preference to frames waiting to be transmitted on the channel. Priority 7 has preference over Priority 0.

**3.35 PNT**: A general reference to phoneline networking transceivers, and especially to the G.995x series of ITU-T Recommendations.

**3.36 preamble**: The fixed signal sequence that is prepended to the physical layer frame. It consists of 4 copies of the TRN sequence.

**3.37 priority slot**: One of 8 slots following the IFG (of a valid transmission or a collision) which are used to implement access priority.

**3.38 QoS contract**: A contract defining a set of negotiated QoS flow parameters between devices involved in a flow. A QoS contract is negotiated between devices at the endpoints of a flow in order to establish buffering and channel (BER/PER) constraints. A QoS contract is negotiated between flow source device and master in order to constrain bandwidth, latency and jitter requirements.

**3.39 registration**: The process used by a G.9954 network device to inform the active network master of its existence and its intention to negotiate future QoS contracts.

**3.40 SMAC mode**: The media access mode used in a master-controlled network.

**3.41 system margin**: A set of values for impairment levels at which a receiver does not exceed a specified frame error rate on a given test loop.

**3.42 transmission opportunity**: An interval of media time, with distinct start-time and length relative to the start of the MAP that can be used by a PNT device for the transmission of frames.

**3.43 TRN16**: The 16-symbol white, constant amplitude QPSK sequence which is used in the physical layer preamble.

**3.44 valid CS frame**: A description of the minimum transmitter signal which should be acceptable to implementations of carrier sense and collision detection.

## 4 Abbreviations and acronyms

This Recommendation uses the following abbreviations:

| | |
|---|---|
| ADSL | Asymmetric Digital Subscriber Line |
| AMAC | Asynchronous MAC Protocol |
| BER | Bit Error Ratio |
| BPS | Bits Per Symbol |
| CBR | Constant Bit Rate |
| CFTXOP | Contention-Free TXOP |
| CR | Collision Resolution |
| CS_IFG | Carrier-Sense IFG |
| CSMA/CD | Carrier Sense Multiple Access with Collision Detection |
| CTXOP | Contention TXOP |
| DFPQ | Distributed Fair Priority Queuing (the G.9951/2 enhanced method for collision resolution (see BEB)) |
| DOCSIS | Data-Over-Cable System Interface Specification |
| FDQAM | Frequency Diverse QAM |
| FEC | Forward Error Correction |
| HCS | Header Check Sequence (a CRC-8 that covers portions of the header and Ethernet address fields) |
| G.9951/2 | Device supporting the G.9951/2 Protocol |
| G.9954 | Device supporting the G.9954 Protocol |
| ICG | Inter-Cycle Gap |
| IFG | Inter-Frame Gap |
| LARQ | Limited Automatic Repeat reQuest (protocol for impulse noise error correction) |
| MAP_IFG | Media Access Plan IFG |
| MII | Media Independent Interface (defined by IEEE Std 802.3 Clause 22) |
| MPDU | MAC Protocol Data Unit |
| NEXT | Near-End crosstalk |
| NID | Network Interface Device (a subscriber line protection device installed at the boundary between the subscriber loop and the in-premise wiring) |
| PAR | Peak to Average Ratio |
| PDU | Protocol Data Unit |
| PE | Payload Encoding |
| POTS | Plain Old Telephone Service (referring to telephony services using the 0-4 kHz spectrum on the phoneline) |
| PNT | Phoneline Networking Transceiver |
| QAM | Quadrature Amplitude Modulation |
| RG | Residential Gateway |

RSVP Resource Reservation Protocol

Self-NEXT Near-End crosstalk from other systems of the same type

SI Scrambler Initialization

SMAC Synchronous MAC protocol

SP Service Provider

TXOP Transmission Opportunity

USB Universal Serial Bus

UTXOP Unallocated TXOP

VBR Variable Bit Rate

# 5 Introduction

## 5.1 G.9954 protocol stack overview

The G.9954 protocol stack is an integrated protocol stack handling PHY, Data Link, Convergence and Management Layers. The G.9954 protocol stack supports both Asynchronous and Synchronous MAC modes of operation. The Asynchronous MAC mode of operation is the same as that defined for the G.9951/2 MAC protocol albeit using a wider range of bauds and constellation encodings. The Synchronous MAC mode is built on top of Asynchronous MAC mode behaviour and represents a functional superset of the Asynchronous MAC mode of operation. This means that a network node running the G.9954 protocol stack inherently knows how to fully function as a native G.9951/2 node.

Synchronous MAC mode depends on the existence of a G.9954 device on the network that is able to assume the role of network master. Such a device is referred to as the "master device" or just "master". A device that is able to assume the role of master on the network is referred to as a master-capable device. A master-capable device is a regular G.9954 device that also supports functional capabilities that allow it to assume the role of master, in the absence of an active master on the network.

The master is responsible for controlling the Synchronous MAC mode of operation by planning media access timing on the network and periodically advertising the media access plan to all devices on the network. The periodic timing is referred to as a MAC cycle. G.9954 nodes operating in Synchronous MAC mode are able to synchronize with the periodic MAC cycle and time their transmissions in accordance with the transmission timing described in the Media Access Plan.

In the presence of a G.9954 master device on the network, G.9954 nodes operate in Synchronous MAC mode; otherwise, they operate in Asynchronous MAC mode. In the presence of G.9951/2 nodes, G.9954 nodes continue to operate in Synchronous MAC mode in the presence of a master; however, they also modify their mode of operation in such a way as to support coexistence and interoperability with G.9951/2 devices operating in Asynchronous MAC mode on the network. This sub-mode is elaborated further in clause 8.

### 5.1.1 Compatibility and interoperability mode

The G.9954 protocol is backward compatible with the G.9951/2 MAC protocol.

Compatibility is provided by conforming to G.9951/2 protocol timing and behaviour and by using a frame format that is backward-compatible with the G.9951/2 frame format. In fact, to a native G.9951/2 node, G.9954 transmissions appear, on the wire, to be indistinguishable from regular G.9951/2 transmissions, albeit possibly at payload bauds that are not supported by the G.9951/2 device.

### 5.1.2 Synchronous MAC mode

The G.9954 MAC protocol is inherently a synchronous MAC protocol that coordinates media access under master control. The protocol is synchronous in the sense that all G.9954 nodes on the network are synchronized to a periodic MAC cycle and transmissions are pre-planned and accurately timed.

The G.9954 synchronous MAC protocol is used to support different kinds of services including asynchronous best-effort data services and isochronous constant and variable bit-rate streaming services such as required by telephony, audio and video.

In a native G.9954 environment, where media access is pre-planned, collision avoidance (CA) strategy is used during normal data-transfer operations. Collision avoidance together with packet aggregation supports more efficient use of the media and provides the infrastructure for supporting Quality of Service guarantees.

The G.9954 synchronous MAC protocol supports bridging to other synchronous protocols, such as IEEE 1394, USB etc. and to Broadband Access protocols such as DOCSIS and IEEE 802.16, using the protocol convergence layer. Furthermore, the master-controlled network model, used in the G.9954 MAC, is a natural model for broadband access networks and is well suited to an architecture containing a Residential Gateway (RG).

### 5.1.3 Quality of Service

The G.9951/2 MAC's support for QoS, based on priority classification using 8 priority levels, provides a basic QoS mechanism for differentiating between different kinds of services. This mechanism is compatible with IEEE 802.1D recommendations and the VLAN Priority Tag (IEEE 802.1P) and the PRECEDENCE bits defined in the original interpretation of the Type Of Service (TOS) field found in an IP packet using the Differentiated Services (Diffserv) protocol.

However, to provide guaranteed QoS, G.9954 provides a mechanism that is compatible with RSVP-like protocols that specifies explicit traffic and rate parameters for a service and not just a relative ordering of packets.

The G.9954 QoS mechanism is based on the concept of a flow, which represents a unidirectional flow of data between network nodes based on well-defined QoS parameters that allow strict control over network throughput, latency, jitter and BER parameters.

Flows are set up and torn down on a service-by-service basis. The G.9954 Link-Layer Control (LLC) and MAC sub-layers are responsible for scheduling the transmission of packets on flows in such a way so as to enforce respective traffic/QoS parameters. Bandwidth is reserved for a flow during its lifetime and this is reflected in the Media Access Plan (MAP) prepared by the master G.9954 node. Bandwidth requirements for a flow may also be modified throughout its lifetime in order to more effectively support changing bandwidth requirements that are characteristic of "bursty" and variable bit-rate (VBR) data streams.

It is the responsibility of the convergence sub-layer to map incoming data streams onto an appropriate flow in order to meet QoS requirements.

Flows may be set up automatically upon service invocation or they may be established at initialization time according to a predefined specification (e.g., part of the convergence layer) or configuration data. Flows may similarly be torn down automatically upon detection of inactivity in order to free network resources associated with the flow.

### 5.1.4 Performance

The G.9954 protocol improves on the performance of the G.9951/2 protocol by avoiding collisions (no collision resolution cycle) and by supporting aggregation of multiple MAC Protocol Data Units (MPDUs) into a single PHY layer burst (frame).

The above performance gains are related to the G.9954 MAC protocol itself and further performance gains and advantages may be expected in implementations themselves.

### 5.1.5 External interfaces and protocols

The G.9954 protocol supports interfaces and bridging to external protocols through the convergence sub-layer in the protocol stack. The particular protocol convergence sub-layer to use is defined either through a configuration parameter or directly through the management interface.

It is the responsibility of the protocol convergence sub-layer to map data packets arriving from a particular interface onto the *flows* appropriate for the particular data service.

Flows defined for particular convergence sub-layer may be set up upon device registration or upon demand. The flow traffic and rate parameters may be predefined for the protocol or may be defined as a configuration parameter in non-volatile storage. Similarly, flows may be set up and configured by management operations performed by an external or remote host.

The G.9954 protocol stack does not assume the existence of an external host processor and is able to directly interface, in hardware, to an external chip, possibly running a different protocol. In this configuration, the protocol convergence sub-layer is assumed to co-reside with the MAC and Link Layers in an integrated G.9954 chip. Alternatively, the G.9954 convergence sub-layer may actually execute, in part or entirely, on an external host processor. The convergence sub-layer may also include address bridging functions/tables.

The external protocols addressed in the G.9954 convergence sub-layer description include the IEEE 802.3/Ethernet protocols and Internet Protocol (IP). In addition, convergence layer support for USB and IEEE 1394 are considered important candidate protocols for transport over G.9954. Furthermore, interfaces to broadband access protocols, such as DOCSIS and possibly even wireless access protocols, such as IEEE 802.11 and IEEE 802.16 are also considered of importance. MPEG video transport streams are assumed to be transported over IP/Ethernet or as MPEG-TS (transport streams).

Protocol mapping and convergence at an explicit level of the protocol stack enables degrees synchronization between external and home networks. This is described in more detail in Appendix III. Furthermore, given QoS defined in terms that are similar to those of the external network, this further supports the extension of QoS methods from external networks into the home network.

### 5.1.6 Security and privacy

A G.9954 network node must register with the master in order to connect to the network and to initiate data transfer. This process of network admission may be extended to provide a basic authorization and privacy support.

A device authorization list may be in the master's configuration file or in some external database accessible from the external host side. This authorization list may define which devices are able to connect to the master and gain access to the network and its resources. Device identification is performed using the device's hardware MAC address. Network access may be denied on the basis of authorization information accessible to the master.

Privacy of data may be supported using encryption methods. Following authorization, the master may run a key management protocol in order to distribute an encryption key to network nodes. Encryption keys are transmitted over the network in an encrypted form using public key encryption methods. Each packet sent on the media, except for certain management packets, can subsequently be encrypted using the shared encryption key.

Privacy may be required in the home in order to protect home content from exposure to unauthorized collection and monitoring caused by cross-talk. Encryption can be used to protect data

in G.9954 transmissions rather than relying on security mechanisms in G.9951/2 based on restricting receiver sensitivity.

Both authentication and privacy support are future and optional features.

### 5.1.7 Management support

Given a home network model based on a Residential Gateway that provides access to services delivered to the home by a service provider, the need to be able to manage, configure, monitor and troubleshoot home networks becomes of increasing importance.

To support this functionality, G.9954 devices are required to support the following management functions:

- Configure, control and monitor all G.9954 devices on network;
- Providing local and remote access to all devices;
- Access to all devices through Gateway (Master);
- Using a message-based protocol (based on G.9951/2 Certification and Diagnostics Protocol);
- Support a standard MIB-structure;
- Support higher-level management interfaces (e.g., SNMP, HTTP, etc.).

The Management facilities are intended to provide access to the following management information:

- PHY information;
- Network information;
- QoS information and statistics;
- Device information;
- Configuration information;
- Authorization and security Information;
- Version information.

## 5.2 Network reference model

This Recommendation defines base-level PHY, MAC, LINK and CONVERGENCE layer functionality.

The primary interface specified is the wire-side electrical and logical interface (W1) between a G.9954 station and the phone wire as shown in Figure 5-1. This Recommendation defines host-side interfaces in terms of example interfaces such as IEEE Std 802.3 logical link level frame formats, addressing and broadcast/multicast behaviour. Several options exist for host-side interfaces, with the MII interface recommendation described in Appendix II.

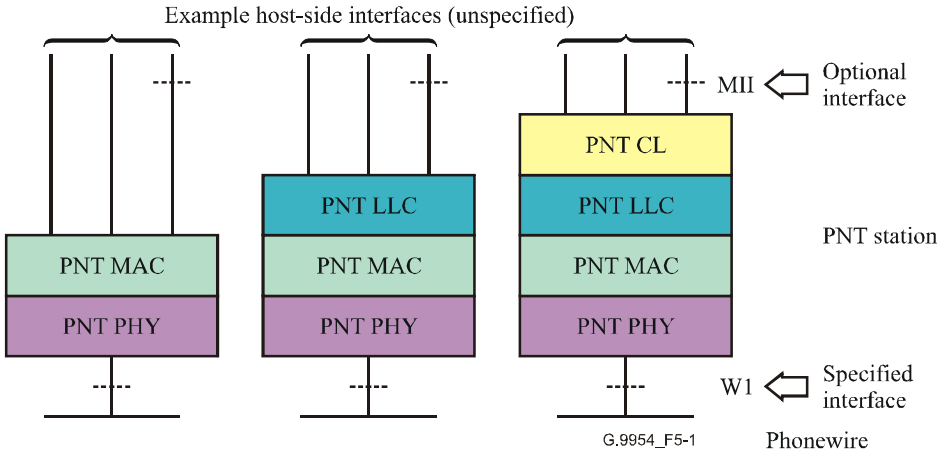


**Figure 5-1/G.9954 – Interfaces**

The PNT system implements a *shared medium* single-segment network, as shown in Figure 5-2. All stations on a segment are logically connected to the same shared channel on the phonewire. Multiple PNT network segments and other network links can be connected through ISO network Layer 2 (L2 or Data Link) or Layer 3 (L3 or IP) relays. Layer 1 relays (PHY layer repeaters) are not defined in this Recommendation.

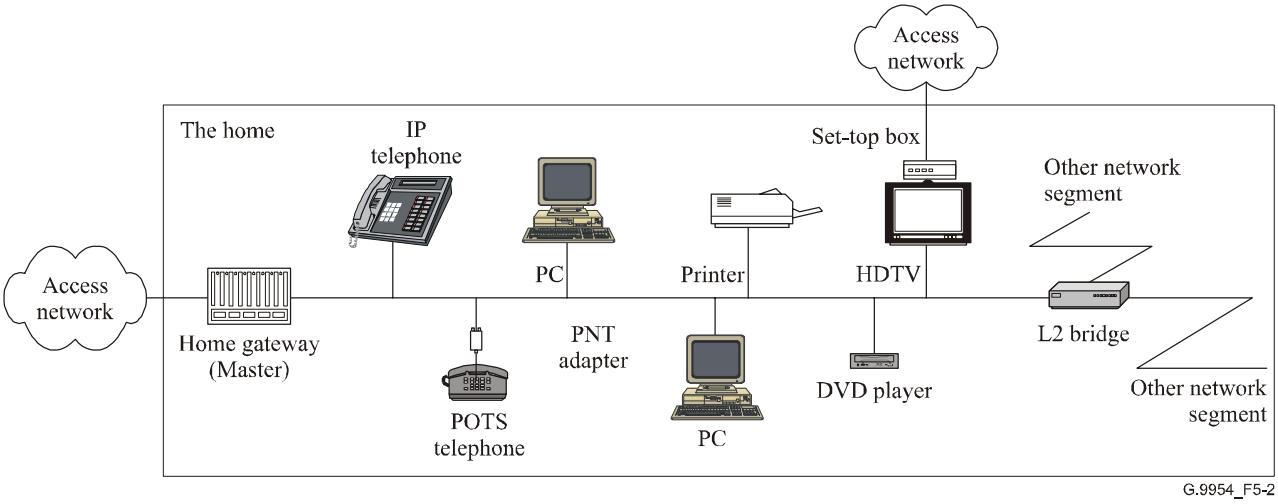


**Figure 5-2/G.9954 – PNT shared medium network segment**

As seen in Figure 5-2, the G.9954 network model assumes a home network composed of a variety of types of network devices, connected to the shared media home phone-line network backbone. It furthermore assumes single or multiple broadband connections to external access networks, through one or more gateway devices and possible bridges to other home network segments, possibly based on other home networking technologies (e.g., wireless, power-line, cable, etc.).

Table 5-1 shows the PNT standard relationship to the ISO/IEC Open Systems Interconnection reference model.

**Table 5-1/G.9954 – Relationship to ISO/IEC open systems interconnection**

| OSI reference model layers | G.9954 layers | |
|---|---|---|
| Application | | |
| Presentation | | |
| Session | | |
| Transport | | |
| Network | | |
| Data link | | Convergence layer |
| | | Link layer protocols |
| | | MAC – Media access control |
| Physical | | PHY |

The PNT network standard is designed to work over "as is" customer premises wiring. The topologies anticipated are random combinations of star, tree and multipoint bus wiring; see Figure 5-3 for an example. Here the "Plain Old Telephone Service" (POTS) Network Interface Device (NID) is shown with the outside subscriber loop to the left, and the premises wiring splitting in a "star" from the NID to several wiring runs. Each run may have one or more modular connectors at wall plates, and variable length *extension* wires (shown as double lines) run from the wall plates to the attached POTS or PNT device. In the example, stations A and B are on one bus; station C is on a second bus, which is unterminated at the end; station E is at the end of a direct run from the NID; and stations F and G share a single wall plate via a two outlet adapter. Many other topologies are possible.

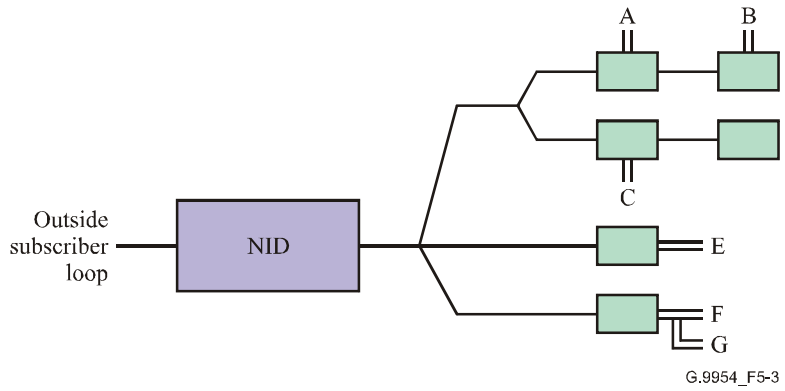


**Figure 5-3/G.9954 – Reference wiring topology**

The G.9954 protocol supports both a master-controlled and master-less network model. The master-controlled network model assumes the existence of a master node that provides the timing on the network and synchronizes media access to all G.9954 network devices. In the absence of a master, the network is considered master-less. The master-less network model is that of G.9951/2 and assumes only a shared-media with devices performing media access using (priority-based) CSMA/CD and collision resolution techniques.

Although media access in a master-controlled network is controlled by the master, communication between two devices does not traverse the master – rather, devices communicate directly

(peer-to-peer) at the master designated time. Any device on the network can potentially act as the master, although it is a role most naturally assumed by a gateway or server device.

## 5.3 The protocol stack

The G.9954 protocol stack provides layer 1 (PHY) and layer 2 (Data-Link) services for transmitting and receiving packets over a wired media using the G.9951/2 and G.9954 protocols. The protocol stack used by the G.9954 is illustrated in Figure 5-4.

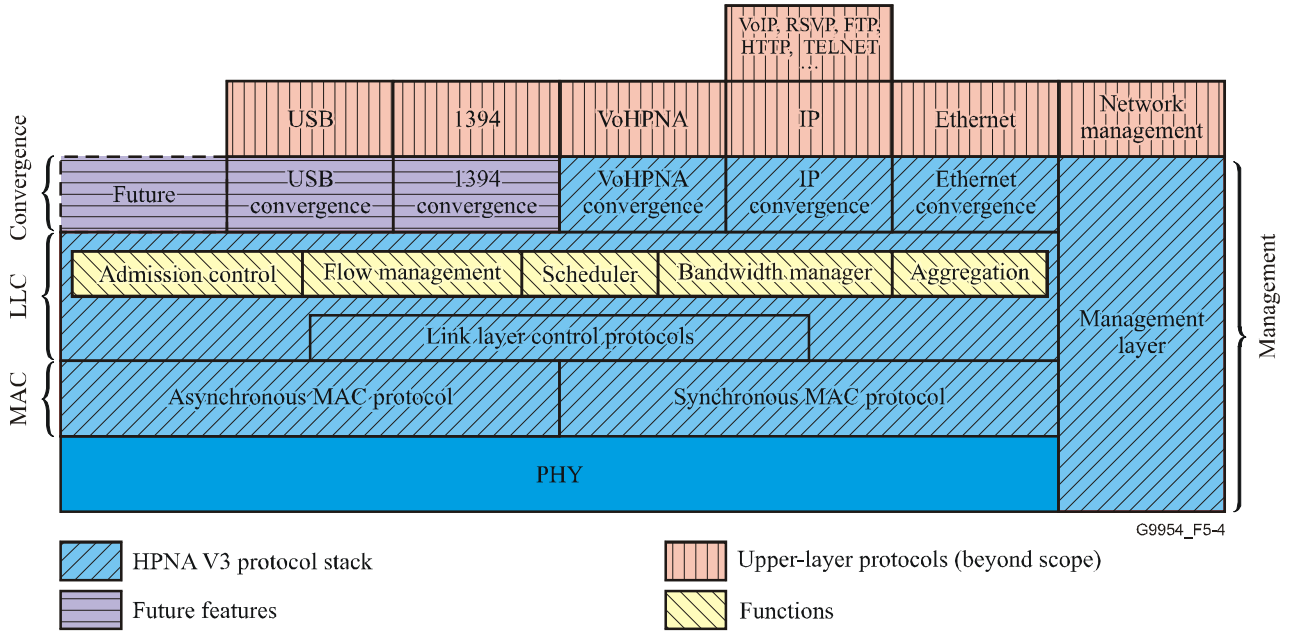


**Figure 5-4/G.9954 – G.9954 protocol stack**

### 5.3.1 PHY layer

The PHY layer provides transmission and reception of physical layer frames using QAM and FDQAM modulation techniques over phone wire media. It supports 2, 4, 8, 16 and 24 Mbaud symbol rates with 2 to 10 bits-per-symbol constellation encoding. This provides a PHY layer data rate in the range of 4-240 Mbit/s within an extended 4-28 MHz PSD mask supporting up to a 24-MHz bandwidth.

### 5.3.2 Data link layer

The Data Link Layer is composed of three sub-layers – the MAC, LLC, and Convergence layers.

#### 5.3.2.1 G.9954 MAC sub-layer

The MAC sub-layer is responsible for managing access to the physical media using a Media Access protocol. It uses the PHY layer to schedule the transmission of MAC Protocol Data Units (MPDUs) over the physical media within PHY layer transport frames.

The G.9954 MAC sub-layer supports media access according to two different protocol modes – Asynchronous mode and Synchronous mode. It supports an asynchronous mode of operation, similar to the G.9951/2 MAC protocol and a synchronous mode of operation as defined in this Recommendation. The asynchronous MAC protocol provides priority-based media access that uses CSMA/CD techniques to control media access and a signalling protocol to resolve media collisions. In contrast, the G.9954 synchronous MAC protocol uses CSMA/CA techniques, under master control, to avoid collisions by pre-planning the timing of all media access.

The G.9954 synchronous mode MAC maintains a vector defining the media access timing planned by the master. Media access timing is planned according to Quality of Service (QoS) constraints for required network services and the plan is broadcast periodically to all G.9954 nodes. G.9954 MACs are responsible for guaranteeing that all media access is performed according to the plan by restricting transmissions only to transmission opportunities (TXOPs) allocated explicitly to it (or to its services) by the master or allocated to a group that it belongs to. The G.9954 master plans media access down to the service level and as such, a G.9954 MAC may schedule packets entirely using the master plan. Alternatively, a G.9954 MAC is allowed to exercise some local QoS intelligence by making scheduling decisions itself within the constraints of the transmission opportunities (TXOPs) allocated to it in the MAP.

The MAC sub-layer is further responsible for providing control information to the PHY layer in order to control the physical characteristics of the transmitted data.

#### 5.3.2.2 G.9954 LLC sub-layer

The LLC sub-layer is responsible for performing link control functions. In particular, it is responsible for managing information concerning network connections, for enforcing Quality of Service (QoS) constraints defined for the various system data flows and for ensuring robust data transmission using Rate Negotiation, Reed-Solomon coding techniques and ARQ (Automatic Repeat ReQuest) techniques.

In addition, the G.9954 Synchronous MAC protocol requires the support of additional Link Control Protocols that manage Network Admission and Flow Setup and Teardown procedures. These protocols are used to manage the information about connected devices and their associated service flows. The link layer protocols interact with upper convergence protocol layers in order to signal such events as device registration, timing events and flow control operations.

In addition to the Link Layer Control Protocols required by the G.9954 Synchronous MAC, the following Link Layer functions are required: Scheduling, Bandwidth Management, Flow Management, Network Admission and Packet Aggregation.

Packet aggregation is used to concatenate multiple MPDUs, within a single PHY layer frame. This concatenation technique is used to increase the size of the PHY frame in order to reduce the overall per-packet protocol overhead. However, the degree of aggregation performed is a function of the latency requirements of services and the size of the allocated transmission opportunity. The LLC sub-layer is responsible for performing this framing and de-framing and for maximizing the size of a burst within the constraints defined by the media access plan.

#### 5.3.2.3 Convergence layer

The convergence layer is a protocol-specific set of sub-layers that map various transport layer protocols into the native primitives of the LLC sub-layer. The LLC sub-layer provides a protocol independent interface and a well-defined QoS framework. It is the responsibility of the convergence sub-layer to translate the native protocol into this underlying framework.

The convergence sub-layer may use protocol or configuration specific information to perform the translation.

#### 5.3.2.4 Management layer

The management layer described in the protocol stack in Figure 5-4 includes both Network Layer Management and G.9954 Management facilities. Network Layer Management operates on Network and Transport Layers, using higher level management protocols such as SNMP and as such are beyond the scope of this Recommendation.

G.9954 management includes all those facilities that are required in order to collect information from the PHY, MAC, Link and Convergence layers of the G.9954 device or remote devices and to exercise control over them. G.9954 management supports both local and remote management

capabilities. This means that management operations may be performed from a local host interfacing to the G.9954 device from the host side or from a management entity interfacing with the G.9954 device from the network (wire) side using a peer management protocol.

G.9954 device may similarly be configured locally or remotely. Local configuration is performed by either reading configuration settings from non-volatile store or under the control of a local host. Remote configuration may be performed using the remote management protocol, as described above, or downloaded by the master during the network admission procedure.

# 6 PHY layer specification

## 6.1 Overview

The G.9954 PHY layer is an extension of the G.9951/2 PHY layer, supporting three spectral masks and seven bauds, allowing 10 spectral mask/baud combinations:

- Spectral Mask #1: 4-10 MHz; 2, 4 MBaud (same as in G.9951/2);
- Spectral Mask #2: 4-21 MHz; 2, 4, 8, 16 MBaud;
- Spectral Mask #3: 4-28 MHz; 2, 6, 12, 24 MBaud.

Constellation sizes range from 2 to 10 bits per symbol, specifying PHY layer payload modulation rates that range from 4 Mbit/s to 240 Mbit/s.

Information is transmitted on the channel in bursts. Each burst or physical layer frame consists of PHY-layer payload information encapsulated with PHY preamble, header and postamble. The PHY-layer payload refers to the portion of the Link Level Frame that is modulated at the payload rate, which is typically higher than the header rate. Hereafter, "payload" refers to the PHY-layer payload unless otherwise specified.

The following describes the physical layer formatting.

## 6.2 Transmitter reference model

The transmitter block diagram is shown in Figure 6-1. This consists of a frame processor, data scrambler, bit-to-symbol mapper, and QAM modulator, as defined in the following clauses.

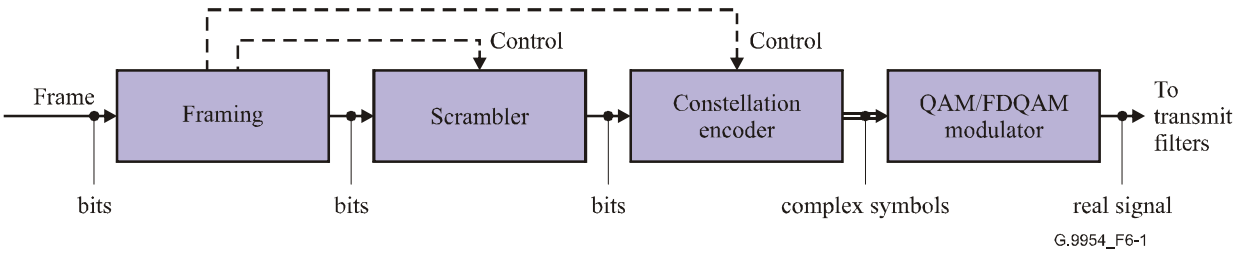


**Figure 6-1/G.9954 – Transmitter block diagram**

## 6.3 Framing

The frame format is shown in Figure 6-2. This consists of a low-rate header section, a variable-rate payload section, and a low-rate trailer. Some parts of the frame are not scrambled, as described in 6.4.

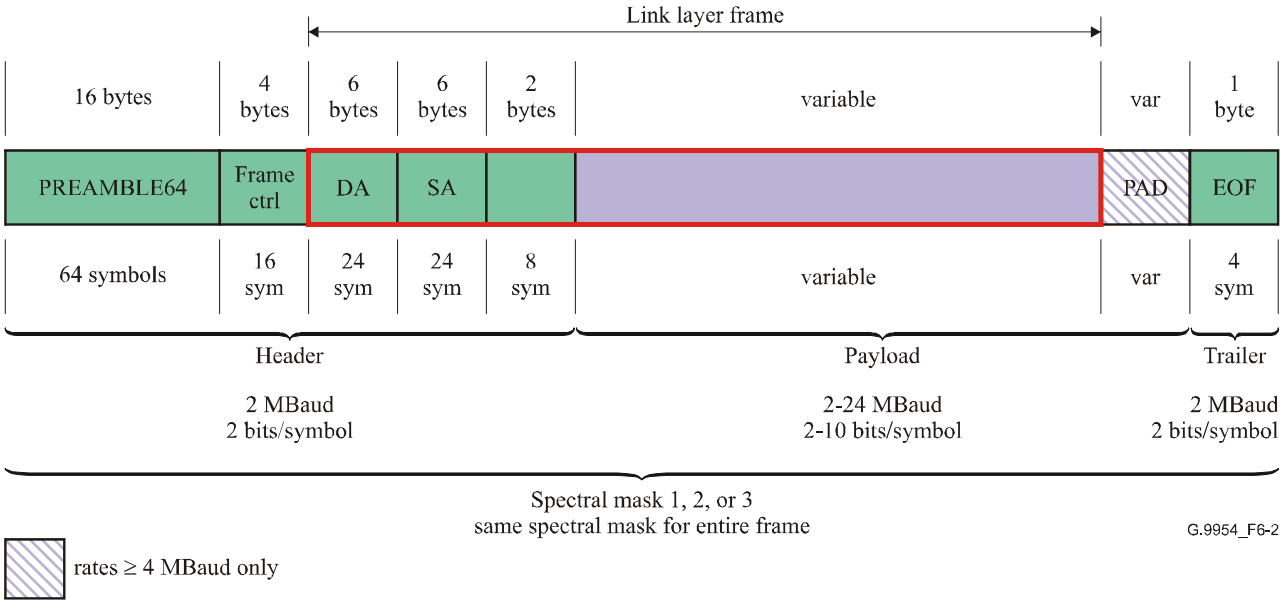


**Figure 6-2/G.9954 – PHY frame format**

The interpretation of the two-byte field following the SA and the variable-length field following it is given by the Link Layer Frame format defined in clause 10.

### 6.3.1 Bit order

Except where otherwise stated, all fields are encoded most significant octet first, least significant bit first within each octet. Bit number 0 is the lsb within a field. Diagrams show MSB bits or octets to the left.

### 6.3.2 Preamble definition

The PREAMBLE64 is defined as a repetition of four 16-symbol sequences (TRN16) that result from encoding 0xfc483084 (in the order defined in 6.3.1) at 2 MBaud, 2 bits-per-symbol, with the scrambler disabled.

NOTE – The TRN16 is a white, constant amplitude QPSK sequence. The preamble was designed to facilitate:

- power estimation and gain control;
- baud offset estimation;
- equalizer training;
- carrier sense;
- collision detection.

### 6.3.3 Frame control definition

The frame control field is a 32-bit field defined in Table 6-1.

**Table 6-1/G.9954 – Frame control fields**

| Field | Bit Number | Bits | Description |
|---|---|---|---|
| FT | 31:24 | 8 | Frame Type.<br>0 Asynchronous MAC frame<br>0x01-0x7F Reserved<br>0x80-0xFE Synchronous MAC frames<br>0xFF Reserved<br>(Bits 31:24 decoded as described below) |
| SMAC | 31:31 | 1 | Synchronous MAC frame |
| FS | 30:28 | 3 | Frame Subtype.<br>0 Ethernet Frame<br>1 MAP (FT = 0x90)<br>2-7 Reserved for future use. |
| RSVD | 27:24 | 4 | Reserved. This field shall be set to zero by the transmitter and the receiver shall discard frames with non-zero values. |
| FID/PRI | 23:20 | 4 | Flow Identifier/Priority<br>If SMAC = 1, bits 23:20 are the Flow ID.<br>If SMAC = 0, bits 22:20 are the Priority (0-7), bit 23 shall be set to zero by the transmitter and shall be ignored by the receiver. |
| SI | 19:16 | 4 | Scrambler Initialization |
| PE | 15:8 | 8 | Payload Encoding |
| HCS | 7:0 | 8 | Header Check Sequence |

Hence, with the bit-ordering defined in 6.3.1, the frame control fields are transmitted in the order shown in Figure 6-3.

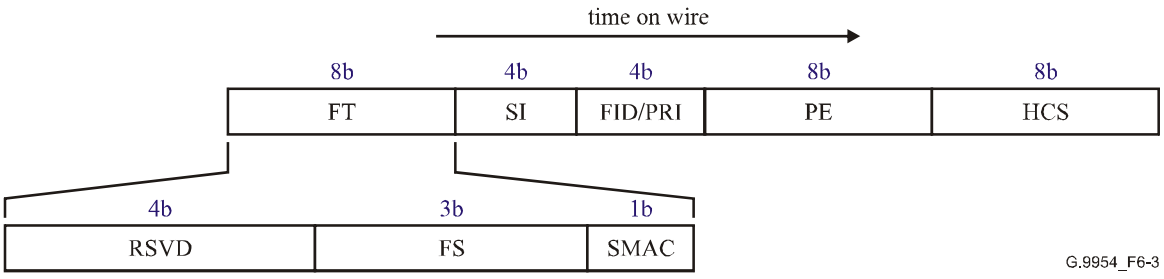


**Figure 6-3/G.9954 – Frame control field order**

#### 6.3.3.1 Frame Type

The Frame Type (FT) is an eight-bit field that is used to define different frame formats. G.9951/2 devices shall transmit 0 in this field, and shall discard any frames with FT other than zero.

G.9954 devices may transmit frames with FT = 0, 0x80 or 0x90. All other values are reserved.

The FT is intended to provide a mechanism for Forward Compatibility, allowing extensions to use frame formats differing from G.9951/2 or G.9954.

The FT field is composed of sub-fields as follows:

#### 6.3.3.1.1 Frame SubType (FS)

This field is used to define the Frame Subtype. The following subtypes are defined:

- 0 = Ethernet frame
- 1 = MAP frame used in synchronous MAC protocol
- 2-7 = Reserved for future use.

Reserved frame subtype values are intended for use in future versions to support frame types associated with other Convergence layers.

#### 6.3.3.1.2 Synchronous MAC (SMAC)

This bit-field is used to indicate a synchronous MAC frame. It is used to define the interpretation of the overloaded PRI/FID fields. The SMAC bit shall not be set in frames sent to G.9951/2 devices or in broadcast/multicast frames in the presence of G.9951/2 devices.

#### 6.3.3.1.3 Reserved bits (RSVD)

This field shall be set to zero by the transmitter, and the receiver shall discard any frame where this field is non-zero.

### 6.3.3.2 Scrambler Initialization Bits

This 4-bit field shall be set to the value used to initialize the scrambler, as described in 6.4.

### 6.3.3.3 FID/PRI

The interpretation of this field is dependent on the value of the SMAC field. When SMAC = 0, the interpretation of the field is the PHY-level priority for transmission and class-of-service priority indication for the receiver. When SMAC = 1, the interpretation of the field is the class-of-service Flow ID associated with the frame.

#### 6.3.3.3.1 Priority

Priority refers to the media access priority mechanism, see the Media Access Protocol Specification in clause 7. The 3-bit PHY priority value (PRI) refers to the absolute priority that a given frame will be given when determining media access, and is the value used in the PNT MAC. Priority 7 frames have preferential access over Priority 0.

PRI is a field carried in the PHY-level frame transmission and is intended to indicate a 3-bit PHY-level priority or class-of-service indication to the receiver link level processor for managing priority and class of service of the received frame. The PRI value is not used by the receiver PHY processor.

For stations that do not implement class-of-service the PRI field shall be ignored on receive, and shall be transmitted with a value set to PRI = 2.

See the Media Access Protocol Specification – Asynchronous MAC Mode Operation in clause 7 for a description of how priority values are used.

#### 6.3.3.3.2 Flow ID (FID)

This 4-bit field overloads the Priority (PRI) field and is used to identify the QoS *Flow* associated with the frame. The Flow ID interpretation is used only when the value of the SMAC field is 1. The FID field is used by MAC, Link and Convergence layer processors in both the transmitter and receiver to manage quality-of-service. It is not used by the transmitter and receiver PHY processor.

See the Media Access Protocol Specification – Synchronous MAC Mode Operation in clause 7 for a description of how Flow IDs are used.

#### 6.3.3.4 Payload encoding

This field determines the spectral mask, baud and the constellation encoding of the payload bits. This field is defined by the following sub-fields.

**Table 6-2/G.9954 – Payload encoding fields**

| Field | Bit Number | Bits | Description |
|---|---|---|---|
| EBPS | 7 | 1 | Extended Bits per Symbol |
| SM | 6:5 | 2 | Spectral Mask |
| BAUD | 4:3 | 2 | Symbol Rate |
| BPS | 2:0 | 3 | Bits Per Symbol |

A *band* is defined as a combinations of bauds, modulation type and carrier frequency.

##### 6.3.3.4.1 Extended Bits per Symbol Bit

The EBPS is used to indicate an extended encoding of the BPS field. More specifically, it is used to extend the interpretation of the BPS field when EBPS=1. This is described in detail in 6.3.3.4.4.

##### 6.3.3.4.2 Spectral mask

The values are defined as follows:

**Table 6-3/G.9954 – Spectral mask subfield**

| SM value | Interpretation |
|---|---|
| 0 | Spectral Mask #1 (4-10 MHz) |
| 1 | Spectral Mask #2 (4-21 MHz) |
| 2 | Spectral Mask #3 (4-28 MHz) |
| 3 | Reserved on transmit, discard frame on receive. |

See 6.8.3 for definitions of the spectral masks.

##### 6.3.3.4.3 Symbol rate

For Spectral Mask #1, the values are defined as follows:

**Table 6-4/G.9954 – Symbol rates for Spectral Mask #1**

| Baud value | Interpretation |
|---|---|
| 0 | Symbol rate = 2 MHz |
| 1 | Symbol rate = 4 MHz |
| 2 | Reserved on transmit, discard frame on receive. |
| 3 | Reserved on transmit, discard frame on receive. |

For Spectral Mask #2, the values are defined as follows:

**Table 6-5/G.9954 – Symbol rates for Spectral Mask #2**

| Baud value | Interpretation |
|---|---|
| 0 | Symbol rate = 2 MHz |
| 1 | Symbol rate = 4 MHz |
| 2 | Symbol rate = 8 MHz |
| 3 | Symbol rate = 16 MHz |

For Spectral Mask #3, the values are defined as follows:

**Table 6-6/G.9954 – Symbol rates for Spectral Mask #3**

| Baud value | Interpretation |
|---|---|
| 0 | Symbol rate = 2 MHz |
| 1 | Symbol rate = 6 MHz |
| 2 | Symbol rate = 12 MHz |
| 3 | Symbol rate = 24 MHz |

#### 6.3.3.4.4 Bits per symbol

The values are defined as follows:

**Table 6-7/G.9954 – Bits per symbol encoding**

| EBPS value | BPS value | Interpretation |
|---|---|---|
| 0 | 0 | Reserved on transmit, discard frame on receive |
| 0 | 1 | 2 bits per symbol |
| 0 | 2 | 3 bits per symbol |
| 0 | 3 | 4 bits per symbol |
| 0 | 4 | 5 bits per symbol |
| 0 | 5 | 6 bits per symbol |
| 0 | 6 | 7 bits per symbol |
| 0 | 7 | 8 bits per symbol |
| 1 | 0 | 8-round constellation; 8 bits per symbol |
| 1 | 1 | 9-round constellation; 9 bits per symbol |
| 1 | 2 | 10-round constellation; 10 bits per symbol |
| 1 | 3-7 | Reserved on transmit, discard frame on receive |

### 6.3.3.5 Header Check Sequence (HCS)

An 8-bit cyclic redundancy check (CRC) is computed as a function of the 128-bit sequence in transmission order starting with the FT bits and ending with the Ethernet source address (SA) bits, with zeros substituted for the as-of-yet uncomputed HCS field. The encoding is defined by the following generating polynomial:

$$G(x) = x^8 + x^7 + x^6 + x^4 + x^2 + 1$$

Mathematically, the CRC value corresponding to a given frame is defined by the following procedure.

The first 8 bits of the input bit sequence in transmission order are complemented.

The 128 bits of the sequence in transmission order are then considered to be the coefficients of a polynomial $M(x)$ of degree 127. (The first bit of the FT field corresponds to the $x^{127}$ term and the last bit of the SA field corresponds to the $x^0$ term.)

$M(x)$ is multiplied by $x^8$ and divided by G($x$), producing a remainder $R(x)$ of degree $\leq 7$.

$R(x)$ is multiplied by $H(x)$ to produce $N(x)$, where $H(x)$ is defined as $H(x) = x^7 + x^6 + x^5 + x^4 + x^2 + x + 1$.

$N(x)$ is divided by $G(x)$, producing a remainder $R'(x)$ of degree $\leq 7$.

The coefficients of $R'(x)$ are considered to be an 8-bit sequence.

The bit sequence is complemented and the result is the CRC'.

The 8 bits of the CRC' are placed in the HCS field so that $x^7$ is the least-significant bit of the octet and $x^0$ term is the most-significant bit of the octet. (The bits of the CRC' are thus transmitted in the order $x^7, x^6, .. x^1, x^0$.)

Although the HCS is embedded within the protected bit-stream, it is calculated in such a way that the resulting 128-bit stream provides error-detection capabilities identical to those of a 120-bit stream with an 8-bit CRC appended. The resulting 128-bit sequence, considered as the coefficients of a polynomial of degree 127, when divided by $G(x)$, will always produce a remainder equal to $x^7 + x^6 + x + 1$.

The input bits are unscrambled.

Because all fields covered by the HCS are transmitted at 2 MBaud and 2 bits per symbol (as described in 6.5.1), these fields should be received correctly in many cases where the payload is received in error. The HCS may be used in conjunction with soft-decision error statistics to determine with high probability whether the header was received correctly. This knowledge may be useful for optimizing the performance of ARQ and/or Rate Negotiation algorithms.

### 6.3.4 Link Layer Frame

The bit fields following the frame control field and preceding the pad field are defined in the G.9954 Link Layer specification in clause 10. The first 6 octets are the Destination Address and the next 6 octets are the Source Address.

The presence of the DA and SA in the low-rate header enables reliable error-detection, which is useful for rate selection.

### 6.3.5 Pad

For payloads encoded at rates greater than or equal to 4 MBaud, a variable-length *pad* field consisting of an integer number of octets shall be inserted. The last octet of the pad field (PAD_LENGTH) shall be 255 (0xff) or the number of zero octets (0x00) preceding PAD_LENGTH, whichever is less. The number of zero octets shall ensure that the minimum length of the transmission, from the first symbol of the PREAMBLE64 through the last symbol of the end of frame delimiter, is at least 92.5 μs. For 2-MBaud payloads, there shall not be a pad field.

An example of a compliant formula for generating PAD_LENGTH is:

$$\min\left\{255, \left\lceil \frac{(92.5\,\mu s - 68\,\mu s - 2\,\mu s) \times \mathrm{B}\,\frac{Msymbol}{\text{second}} \times \mathrm{BPS}\,\frac{bit}{symbol}}{8\,\frac{bit}{octet}} \right\rceil - 1 - N\right\}$$

where the baud, B is either 4, 6, 8, 12, 16, or 24, BPS is the bits per symbol, N is the number of octets in the part of the link layer frame transmitted in the payload-rate, 68 µs is the length of the header, and 2 µs is the length of the trailer. If the formula results in a negative quantity, it means that no pad is required.

This ensures that a collision fragment can be discriminated from a valid frame by the transmission length detected by the carrier sense function; see 7.2.

### 6.3.6 End of Frame (EOF) delimiter

The End-of-Frame sequence consists of the first 4 symbols of the TRN sequence, or **0xfc** encoded as 2 bits per symbol at 2 MBaud.

This field is provided to facilitate accurate end-of-carrier-sensing in low-SNR conditions. A station demodulating a frame can use this field to determine exactly where the last payload symbol occurred.

## 6.4 Scrambler

The scrambler is the frame-synchronized scrambler shown in Figure 6-4, which uses the following generating polynomial.

$$G(x) = x^{23} + x^{18} + 1$$

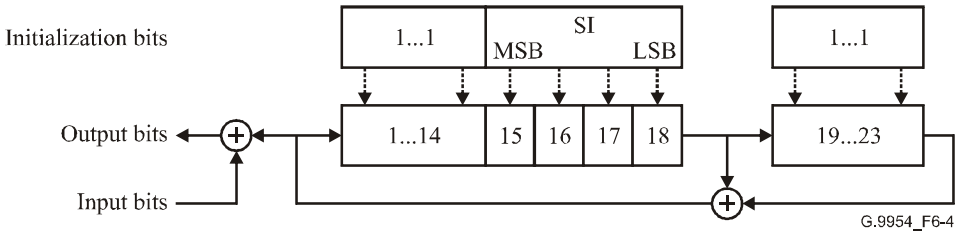


**Figure 6-4/G.9954 – Data scrambler**

Bits 15 through 18 of the shift register shall be initialized with a 4-bit pseudo-random number. This value shall be placed in the SI field defined in 6.3.3.2 in the order such that register position 15 is the MSB (bit 19 of frame control) and bit 18 is the LSB (bit 16 of frame control).

The scrambler shall be bypassed during the preamble bit field and the first 16 bits of Frame Control. The scrambler shall be initialized and enabled starting with the 17th bit of the Frame Control field.

The scrambler shall be bypassed after the last bit of the Link Layer Frame, or the last bit of the PAD field, if present. The EOF sequence shall not be scrambled.

The use of a pseudo-random initial scrambler state results in a more uniform power-spectral density (PSD) measured over multiple similar frames. This eliminates the problem of tones in the PSD from highly correlated successive packets.

## 6.5 Constellation encoder

### 6.5.1 Constellation encoding control

All Header bits up to and including the first two bytes following the SA field shall be encoded at 2 MBaud, 2 bits per symbol. If Spectral Mask #2 or #3 is being used, the output symbols shall be modified as described in 6.5.6.

Starting with the 1st bit following the two bytes following the SA field, the bits shall be encoded according to the PE field (see Table 6-2) up to the last bit of the Link Layer Frame, or the last bit of PAD if it is present.

The EOF sequence shall be encoded at 2 MBaud, 2 bits per symbol. If Spectral Mask #2 or #3 is being used, the output symbols shall be modified as described in 6.5.6.

### 6.5.2 Bit-to-symbol mapping

The incoming bits shall be grouped into N-bit symbols, where N is the number of bits per symbol specified in the PE field. The bit-to-symbol mapping is shown in Figures 6-5 through 6-14. The symbol values are shown with bits ordered such that the right-most bit is the first bit received from the scrambler and the left-most bit is the last bit received from the scrambler.

All constellations except for 3 bits per symbol lie on a uniform square grid, and all constellations are symmetric about the real and imaginary axes.

For the round constellations, only the 1st quadrant is shown and the 2 left-most bits are omitted from the figures. For these cases, the 2 left-most bits are specified in Figure 6-5.

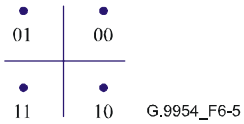


**Figure 6-5/G.9954 – 2 bits per symbol**

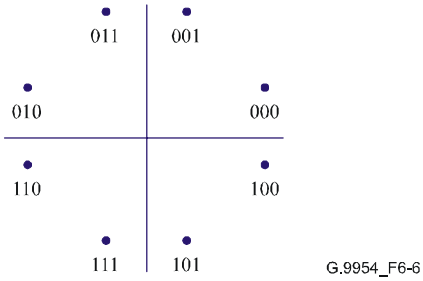


**Figure 6-6/G.9954 – 3 bits per symbol**

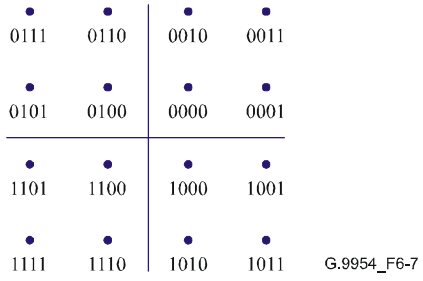


**Figure 6-7/G.9954 – 4 bits per symbol**

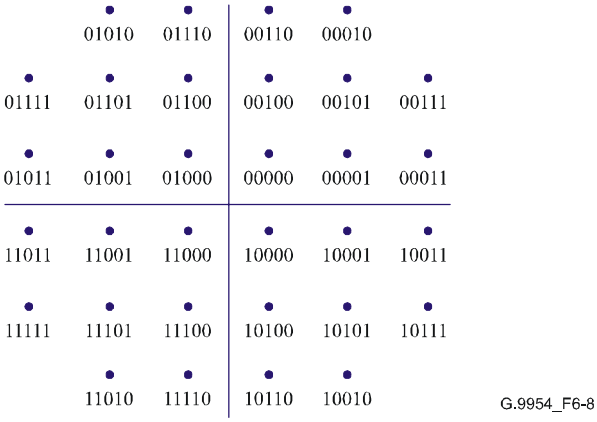


**Figure 6-8/G.9954 – 5 bits per symbol**

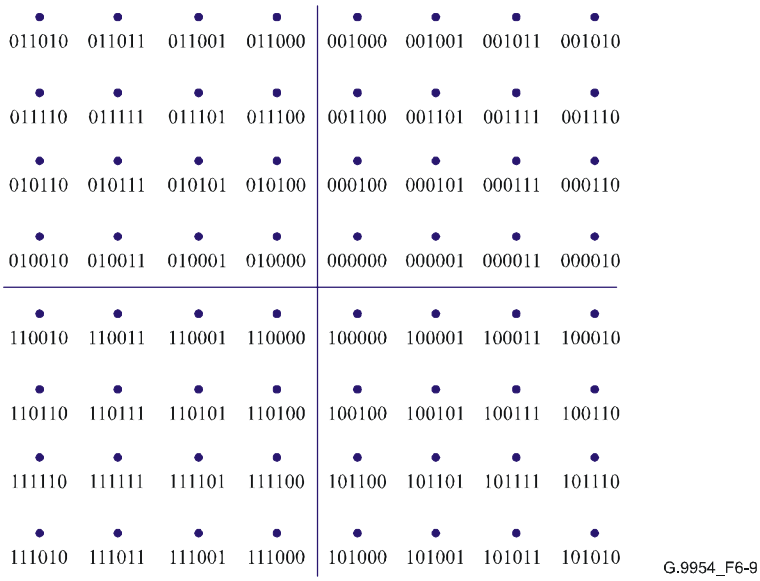


**Figure 6-9/G.9954 – 6 bits per symbol**

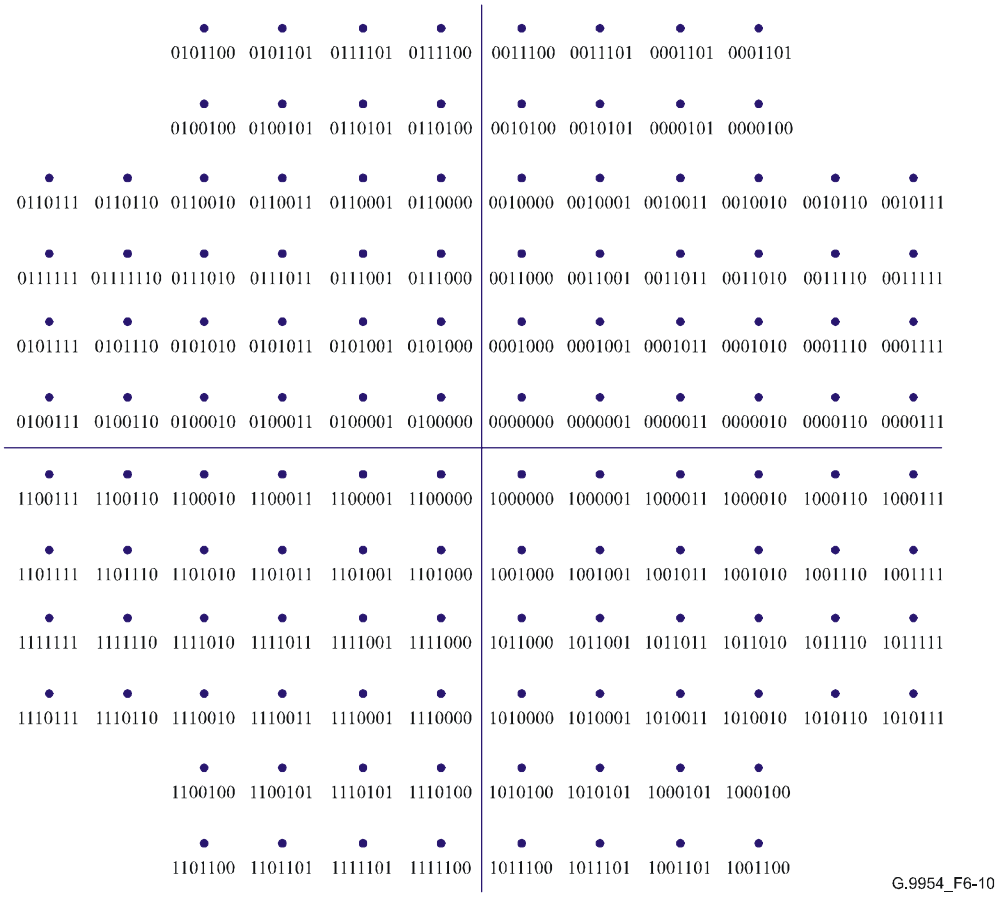


**Figure 6-10/G.9954 – 7 bits per symbol**

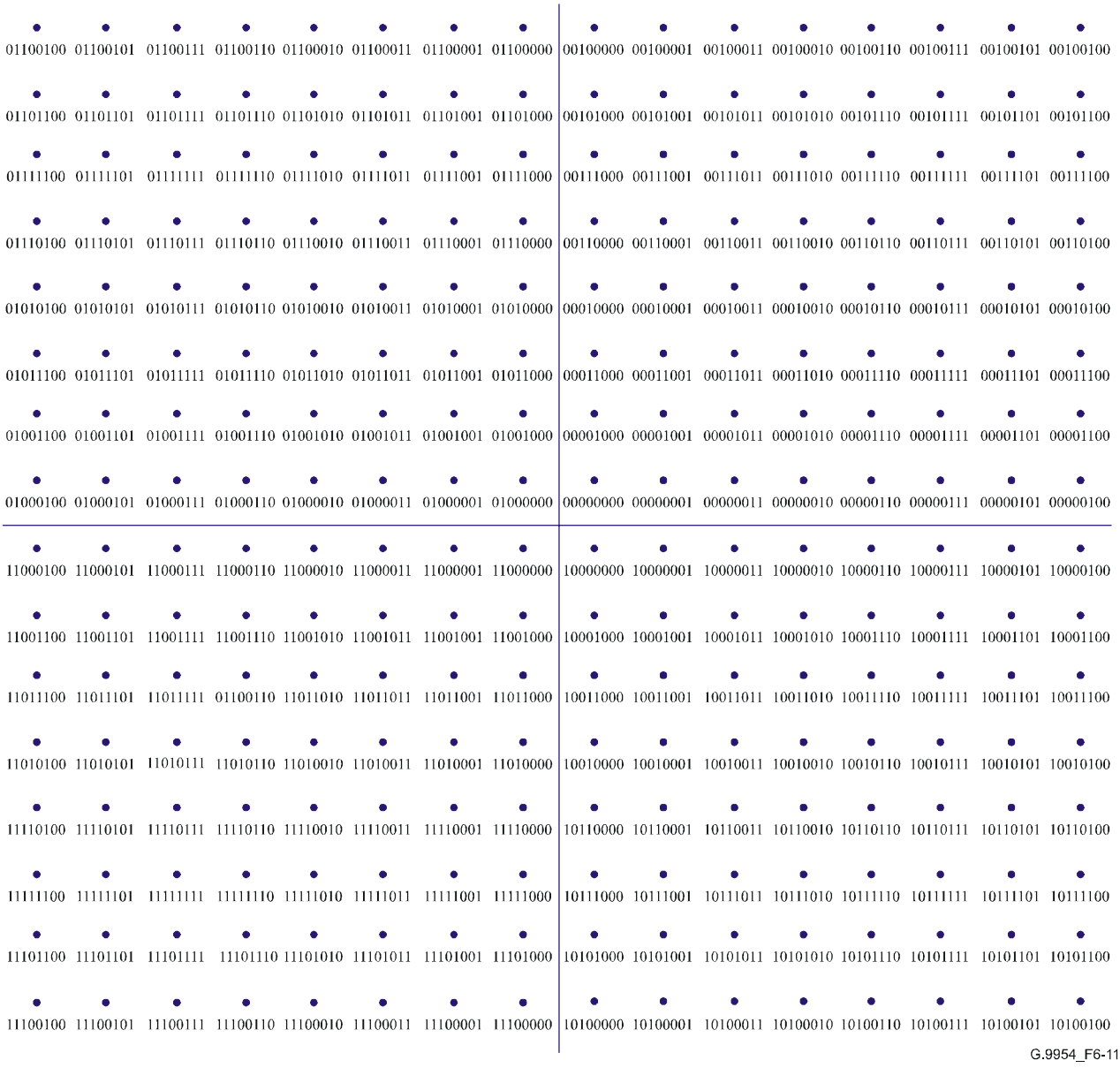


**Figure 6-11/G.9954 – 8 bits per symbol**

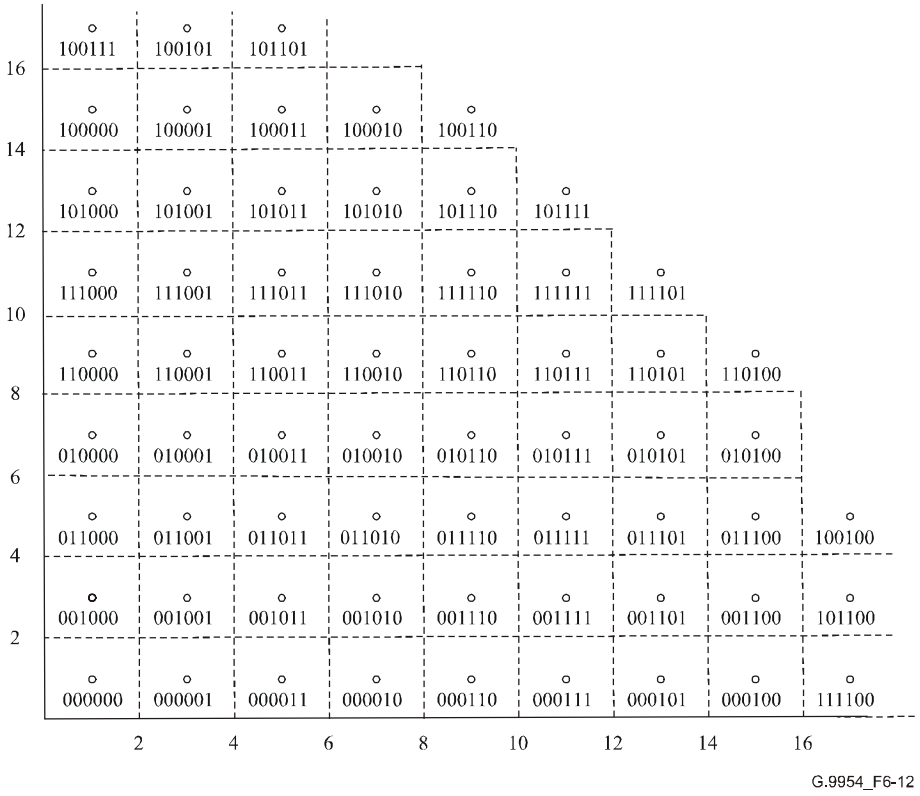


**Figure 6-12/G.9954 – 8 bits per symbol round constellation**

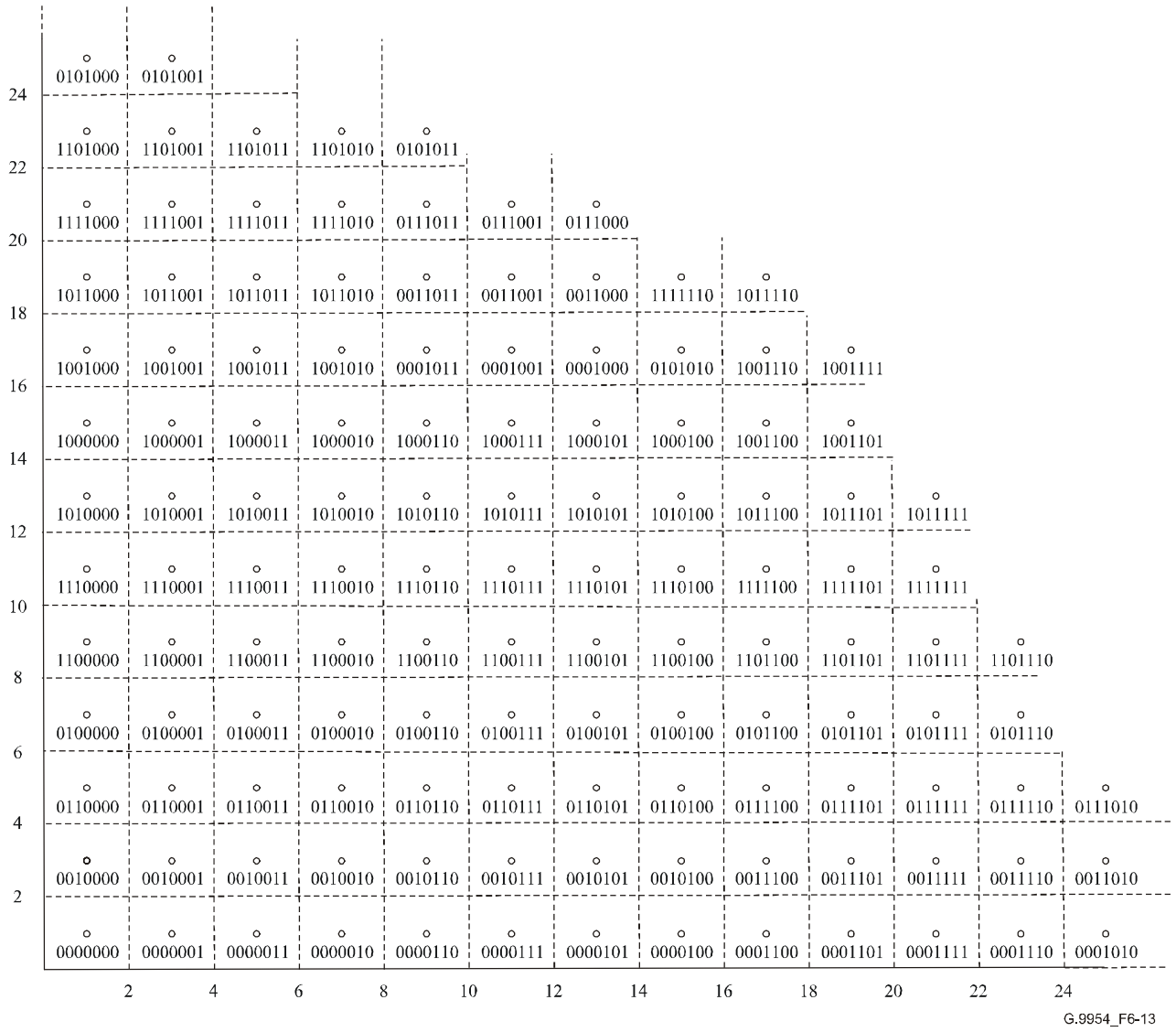


**Figure 6-13/G.9954 – 9 bits per symbol round constellation**

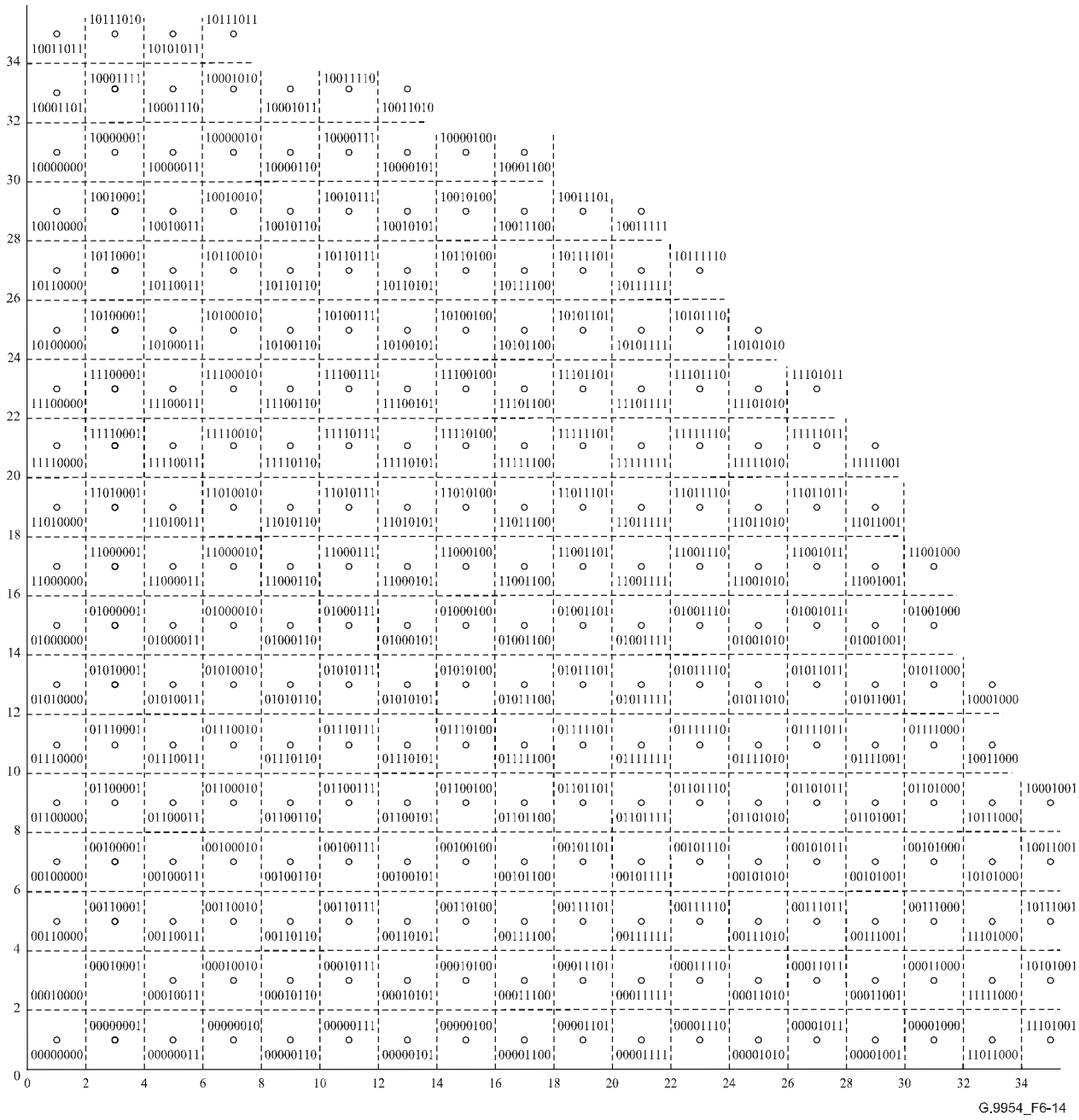


**Figure 6-14/G.9954 – 10 bits per symbol round constellation**

### 6.5.3 Constellation scaling

The relative scaling of different constellations at a single baud is given by Tables 6-8 and 6-9, where the value of s(PE) in Table 6-8 is given by Table 6-9. The value of each constellation point must be accurate to within plus or minus 4 percent of the distance between nearest neighbors in that constellation.

NOTE – For example, at 2 MBaud, 2 bits per symbol, the tolerance on each point is ±0.08, while at 2 MBaud, 5 bits per symbol, the tolerance is ±0.02. Note that the tolerance is not implied by the number of significant digits in Table 6-9; i.e., the values in Table 6-9 should be considered exact.

**Table 6-8/G.9954 – Constellation reference points**

| Bits per symbol | Reference point(s) | Value |
|---|---|---|
| 2 | 00 | (1 + i) × s(PE) |
| 3 | 000 | (12 + 5i) × s(PE) |
| | 001 | (5 + 12i) × s(PE) |
| 4 | 0000 | (1 + i) × s(PE) |
| 5 | 00000 | (1 + i) × s(PE) |
| 6 | 000000 | (1 + i) × s(PE) |
| 7 | 0000000 | (1 + i) × s(PE) |
| 8 | 00000000 | (1 + i) × s(PE) |

**Table 6-9/G.9954 – Constellation scale factors s(PE)**

| Spectral Mask | Symbol rate (MHz) | 2 BPS | 3 BPS | 4 BPS | 5 BPS | 6 BPS | 7 BPS | 8 BPS | 8 BPS round | 9 BPS round | 10 BPS round |
|---|---|---|---|---|---|---|---|---|---|---|---|
| #1 | 2 | 1.0000 | 0.1111 | 0.3333 | 0.2500 | 0.1429 | 0.1111 | 0.0667 | 0.0800 | 0.0556 | 0.0400 |
| | 4 | 0.7071 | 0.0786 | 0.2357 | 0.1768 | 0.1010 | 0.0786 | 0.0471 | 0.0566 | 0.0393 | 0.0283 |
| #2 | 2 | 1.0000 | 0.1111 | 0.3333 | 0.2500 | 0.1429 | 0.1111 | 0.0667 | 0.0800 | 0.0556 | 0.0400 |
| | 4 | 0.7071 | 0.0786 | 0.2509 | 0.1812 | 0.1113 | 0.0835 | 0.0534 | 0.0617 | 0.0431 | 0.0306 |
| | 8 | 0.5000 | 0.0556 | 0.1952 | 0.1396 | 0.0897 | 0.0664 | 0.0438 | 0.0470 | 0.0332 | 0.0235 |
| | 16 | 0.3119 | 0.0335 | 0.1225 | 0.0860 | 0.0583 | 0.0418 | 0.0288 | 0.0296 | 0.0210 | 0.0148 |
| #3 | 2 | 1.0000 | 0.1111 | 0.3333 | 0.2500 | 0.1429 | 0.1111 | 0.0667 | 0.0800 | 0.0556 | 0.0400 |
| | 6 | 0.5774 | 0.0642 | 0.2466 | 0.1664 | 0.1073 | 0.0763 | 0.0512 | 0.0550 | 0.0390 | 0.0275 |
| | 12 | 0.4082 | 0.0454 | 0.1789 | 0.1234 | 0.0816 | 0.0586 | 0.0397 | 0.0419 | 0.0297 | 0.0210 |
| | 24 | 0.2887 | 0.0321 | 0.1185 | 0.0832 | 0.0560 | 0.0404 | 0.0276 | 0.0287 | 0.0202 | 0.0143 |

For Spectral Mask #1, the constellation points are scaled such that the outermost points have approximately equal magnitude. For Spectral Masks #2 and #3, the constellations are scaled based on a statistical measure of the peak to average ratio (PAR).

### 6.5.4 Symbol timing during baud transitions

On a transition from 2 MBaud to a higher baud, the first higher rate symbol shall occur 0.5 microseconds after the last 2-MBaud symbol.

On a transition from a higher baud to 2 MBaud, the first 2-MBaud symbol shall occur 0.5 microseconds after the last higher baud symbol.

For example, the transitions from 2 to 4 MBaud and from 4 to 2 MBaud are illustrated in Figure 6-15.

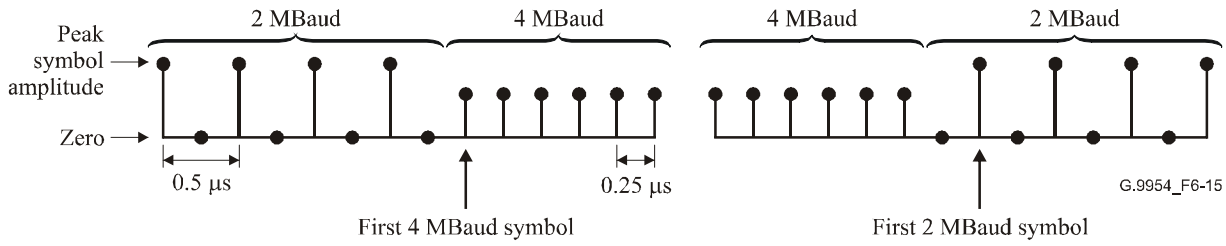


**Figure 6-15/G.9954 – Baud transitions**

### 6.5.5 Encoding rate transitions

If the number of bits in a sequence is not an integer multiple of the number of bits per symbol, then enough zero bits shall be inserted at the end of the bit-stream to complete the last symbol. The number of zero bits inserted shall be the minimum number such that the length of the appended bit stream is an integer multiple of the number of bits per symbol.

### 6.5.6 Modified header and trailer for Spectral Masks #2 and #3

For Spectral Masks #2 and #3, the constellation encoder shall negate every other symbol of the header and trailer starting with the second symbol. That is, symbols 2,4,6 ... 136 of the header and symbols 2 and 4 of the EOF shall be multiplied by –1.

This operation compensates for the new carrier frequencies to create a signal whose header and trailer are compatible with those of Spectral Mask #1, enabling demodulation of the header without knowledge of which spectral mask was used.

## 6.6 QAM/FDQAM modulator

The modulator implements Quadrature Amplitude Modulation (QAM). Figure 6-16 shows an example implementation. The carrier frequencies and transmit filters for each spectral mask do not depend on signalling note (baud).

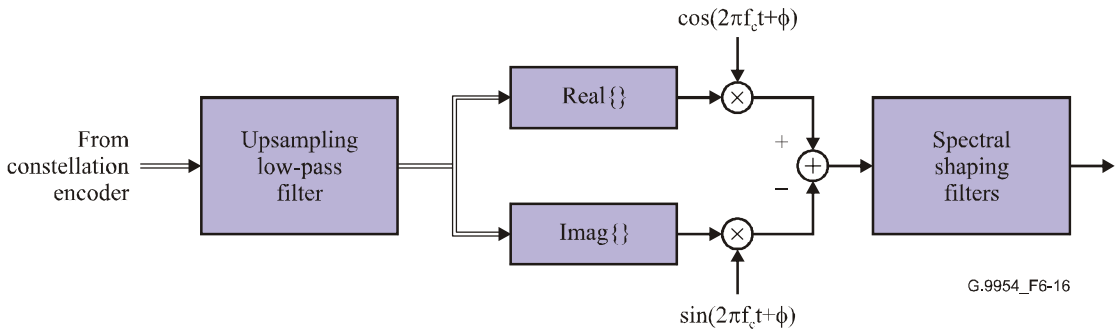


**Figure 6-16/G.9954 – QAM/FDQAM modulator**

### 6.6.1 Carrier frequency and tolerance

Each spectral mask has its own carrier frequency $f_c$:

- Spectral Mask #1: $f_c$ = 7 MHz
- Spectral Mask #2: $f_c$ = 12 MHz
- Spectral Mask #3: $f_c$ = 18 MHz

The carrier clock shall be locked to the symbol clock. So, the carrier frequency tolerance is derived from the clock tolerance defined in 6.9.2.

### 6.6.2 Transmit filters

The details of the transmit filters are implementation dependent. Clauses 6.8.3 and 6.8.4 constrain the transmit filter designs.

## 6.7 Minimum device requirements

Stations at a minimum shall be capable of transmitting and receiving Spectral Mask #1 and Spectral Mask #2. Stations may transmit and receive using Spectral Mask #3.

Stations at a minimum shall be capable of transmitting and receiving 2-, 4-, 8- and 16-MBaud modulated frames.

This implies use of both QAM and FDQAM.

Stations at a minimum shall be capable of transmitting all constellations from 2 bits per symbol to 8 bits per symbol (PE values 1-7) and receiving all constellations from 2 bits per symbol to 6 bits per symbol (PE values 1-5).

## 6.8 Transmitter electrical specification

### 6.8.1 Transmit power

The transmit power shall be between –7 dBm and –9.5 dBm, measured across a 100-ohm load between tip and ring, integrated from 0 to 30 MHz.

### 6.8.2 Transmit voltage

The RMS differential transmit voltage shall not exceed –15 dBVrms in any 2-µs window between 0 and 6 MHz, measured across a 135-ohm load between tip and ring for any payload encoding. The peak differential transmit voltage shall not exceed 580 mVpeak, measured across a 135-ohm load between tip and ring for any payload encoding.

Stations that are not transmitting shall emit less than –65 dBVrms measured across a 100-ohm load between tip and ring.

### 6.8.3 Spectral masks

Three spectral masks are defined. Stations shall transmit using the spectral mask negotiated via the Link Layer Rate Negotiation Control Function.

#### 6.8.3.1 PSD upper bound

When transmitting with Spectral Mask #1, the PNT metallic power spectral density (PSD) shall be constrained by the upper and lower bounds depicted in Figure 6-17 and in Tables 6-10 and 6-11 with the measurement made across a 100-ohm load across tip and ring at the transmitter W1 interface. The upper bound shall apply to all symbol rates and constellations, and the lower bound shall apply to 2 MBaud, 2 bits/symbol.

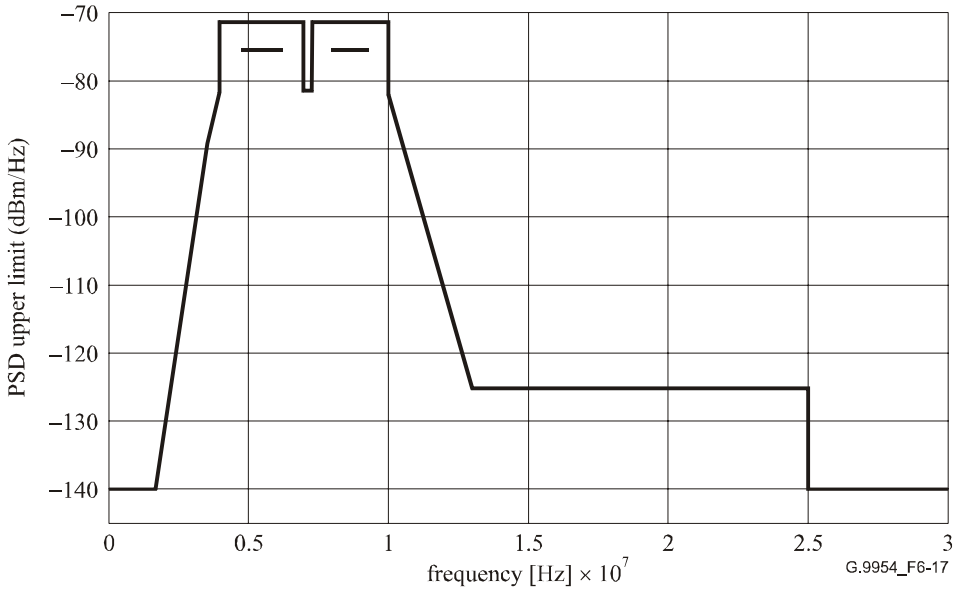


**Figure 6-17/G.9954 – Transmit PSD upper and lower bounds for Spectral Mask #1**

**Table 6-10/G.9954 – Transmit PSD upper bounds for Spectral Mask #1**

| Frequency [MHz] | PSD limit [dBm/Hz] |
|---|---|
| $0.015 < f \leq 1.7$ | –140 |
| $1.7 < f \leq 3.5$ | $-140 + (f - 1.7) \times 50.0/1.8$ |
| $3.5 < f \leq 4.0$ | $-90 + (f - 3.5) \times 17.0$ |
| $4.0 < f < 7.0$ | –71.5 |
| $7.0 \leq f \leq 7.3$ | –81.5 |
| $7.3 < f < 10.0$ | –71.5 |
| $10.0 \leq f < 13.0$ | $-81.5 - (f - 10.0) \times 43.5/3.0$ |
| $13.0 \leq f < 25.0$ | –125 |
| $25.0 \leq f < 30.0$ | –140 |

**Table 6-11/G.9954 – Transmit PSD lower bounds for Spectral Mask #1**

| Frequency [MHz] | PSD limit [dBm/Hz] |
|---|---|
| $4.75 < f < 6.25$ | –76.0 |
| $8.00 < f < 9.25$ | –76.0 |

When transmitting with Spectral Mask #2, the PNT metallic power spectral density (PSD) shall be constrained by the upper bound depicted in Figure 6-18 and in Tables 6-12 and 6-13 with the measurement made across a 100-ohm load across tip and ring at the transmitter W1 interface. The upper bound shall apply to all symbol rates and constellations, and the lower bound shall apply to 2 MBaud, 2 bits/symbol.

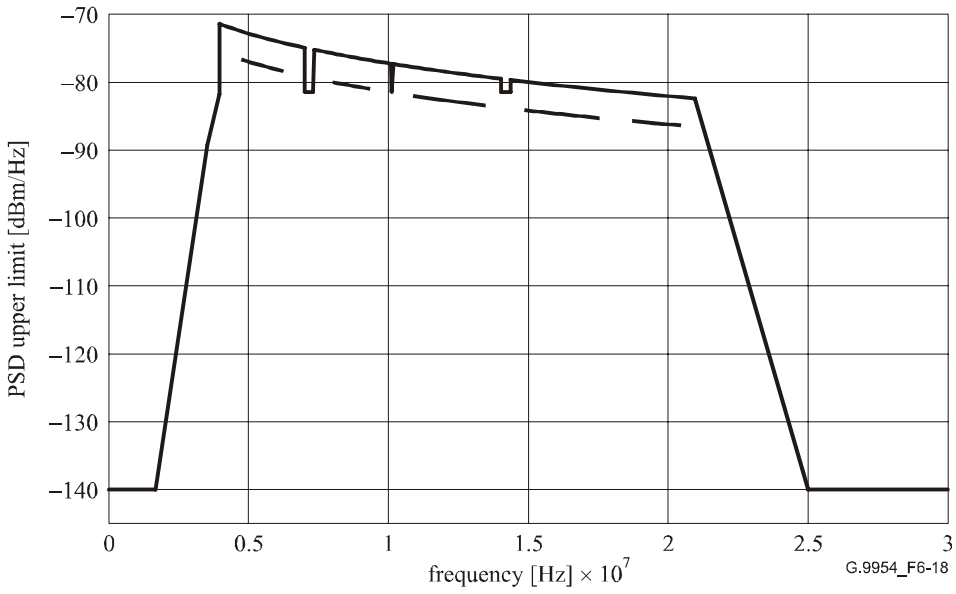


**Figure 6-18/G.9954 – Transmit PSD upper bound for Spectral Mask #2**

**Table 6-12/G.9954 – Transmit PSD upper bound for Spectral Mask #2**

| Frequency [MHz] | PSD limit [dBm/Hz] |
|---|---|
| $0.015 < f \leq 1.7$ | –140 |
| $1.7 < f \leq 3.5$ | $-140 + (f - 1.7) \times 50.0/1.8$ |
| $3.5 < f \leq 4.0$ | $-90 + (f - 3.5) \times 17.0$ |
| $4.0 < f < 7.0$ | $-71.5 - 15 \times \log_{10}(f/4)$ |
| $7.0 \leq f \leq 7.3$ | –81.5 |
| $7.3 < f < 10.1$ | $-71.5 - 15 \times \log_{10}(f/4)$ |
| $10.1 \leq f \leq 10.15$ | –81.5 |
| $10.15 < f < 14.0$ | $-71.5 - 15 \times \log_{10}(f/4)$ |
| $14.0 \leq f \leq 14.35$ | –81.5 |
| $14.35 < f < 18.068$ | $-71.5 - 15 \times \log_{10}(f/4)$ |
| $18.068 \leq f \leq 18.168$ | –81.5 |
| $18.168 < f < 21.0$ | $-71.5 - 15 \times \log_{10}(f/4)$ |
| $21.0 \leq f < 25.0$ | $-82.3 - (f - 21) \times 57.7/4.0$ |
| $25.0 \leq f$ | –140 |

**Table 6-13/G.9954 – Transmit PSD lower bound for Spectral Mask #2**

| Frequency [MHz] | PSD limit [dBm/Hz] |
|---|---|
| $4.75 < f < 6.25$ | $-75.5 - 15 \times \log_{10}(f/4)$ |
| $8.00 < f < 9.35$ | $-75.5 - 15 \times \log_{10}(f/4)$ |
| $10.90 < f < 13.50$ | $-75.5 - 15 \times \log_{10}(f/4)$ |
| $14.85 < f < 17.57$ | $-75.5 - 15 \times \log_{10}(f/4)$ |
| $18.67 < f < 20.25$ | $-75.5 - 15 \times \log_{10}(f/4)$ |

When transmitting with Spectral Mask #3, the PNT metallic power spectral density (PSD) shall be constrained by the upper bound depicted in Figure 6-19, Tables 6-14 and 6-15 with the measurement made across a 100-ohm load across tip and ring at the transmitter W1 interface. The upper bound shall apply to all symbol rates and constellations, and the lower bound shall apply to 2 MBaud, 2 bits/symbol.

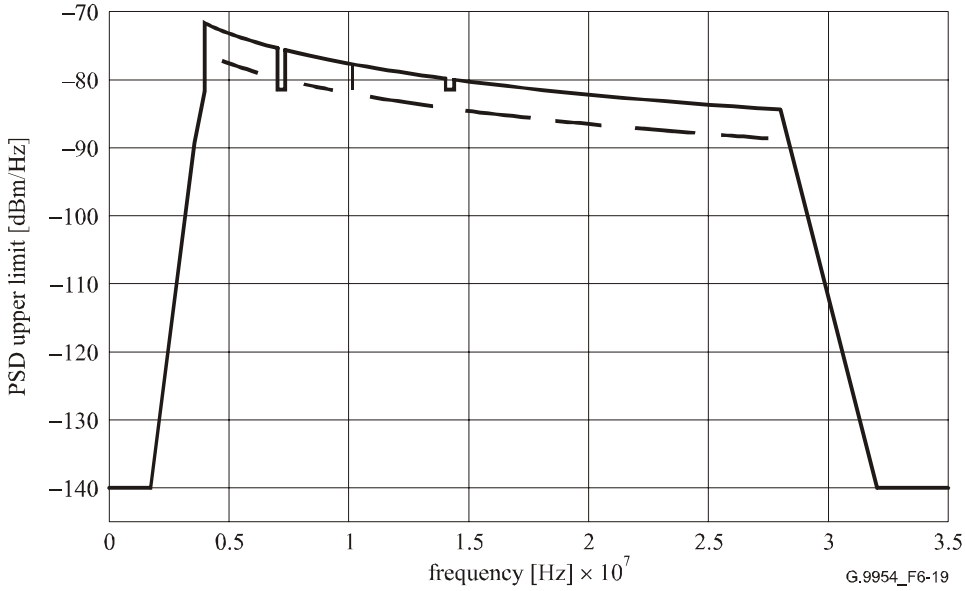


**Figure 6-19/G.9954 – Transmit PSD upper bound for Spectral Mask #3**

**Table 6-14/G.9954 – Transmit PSD upper bound for Spectral Mask #3**

| Frequency [MHz] | PSD limit [dBm/Hz] |
|---|---|
| $0.015 < f \leq 1.7$ | –140 |
| $1.7 < f \leq 3.5$ | $-140 + (f - 1.7) \times 50.0/1.8$ |
| $3.5 < f \leq 4.0$ | $-90 + (f - 3.5) \times 17.0$ |
| $4.0 < f < 7.0$ | $-72.0 - 15 \times \log_{10}(f/4)$ |
| $7.0 \leq f \leq 7.3$ | –81.5 |
| $7.3 < f < 10.1$ | $-72.0 - 15 \times \log_{10}(f/4)$ |
| $10.1 \leq f \leq 10.15$ | –81.5 |
| $10.15 < f < 14.0$ | $-72.0 - 15 \times \log_{10}(f/4)$ |
| $14.0 \leq f \leq 14.35$ | –81.5 |
| $14.35 < f < 28.0$ | $-72.0 - 15 \times \log_{10}(f/4)$ |
| $28 \leq f < 32.0$ | $-84.7 - (f - 28) \times 55.3/4.0$ |
| $32.0 \leq f$ | –140.0 |

**Table 6-15/G.9954 – Transmit PSD lower bound for Spectral Mask #3**

| Frequency [MHz] | PSD limit [dBm/Hz] |
|---|---|
| $4.75 < f < 6.25$ | $-76.0 - 15 \times \log_{10}(f/4)$ |
| $8.00 < f < 9.35$ | $-76.0 - 15 \times \log_{10}(f/4)$ |
| $10.90 < f < 13.50$ | $-76.0 - 15 \times \log_{10}(f/4)$ |
| $14.85 < f < 17.57$ | $-76.0 - 15 \times \log_{10}(f/4)$ |
| $18.67 < f < 20.50$ | $-76.0 - 15 \times \log_{10}(f/4)$ |
| $21.95 < f < 24.40$ | $-76.0 - 15 \times \log_{10}(f/4)$ |
| $25.50 < f < 27.25$ | $-76.0 - 15 \times \log_{10}(f/4)$ |

The resolution bandwidth used to make this measurement shall be 10 kHz for frequencies between 2.0 and 30.0 MHz and 3 kHz for frequencies between 0.015 and 2.0 MHz. An averaging window of 213 seconds shall be used, and 1500-octet MTUs separated by an IFG duration of silence shall be assumed. A total of 50 kHz of possibly non-contiguous bands may exceed the limit line under 2.0 MHz, with no sub-band greater than 20 dB above the limit line. When transmitting with Spectral Mask #1, a total of 100 kHz of possibly non-contiguous bands may exceed the limit line between 13.0 and 30.0 MHz, with no sub-band greater than 20 dB above the limit line. When transmitting with Spectral Mask #2, a total of 100 kHz of possibly non-contiguous bands may exceed the limit line between 25.0 and 30.0 MHz, with no sub-band greater than 20 dB above the limit line.

NOTE 1 – The notches at 4.0, 7.0, 10.1, 14.0, 18.068, 21.0, and 24.9 MHz are designed to reduce RFI egress in the radio amateur bands.

NOTE 2 – The masks should be tested at PE values of 2 and 3 bits/symbol, as these payload encodings result in the maximum transmitted power.

### 6.8.4 Transmitter symbol response

When transmitting with Spectral Mask #1, the symbol response of the transmitter output shall be upper-bounded by the temporal mask shown in Figure 6-20. When transmitting with Spectral Mask #2 or Spectral Mask #3, the symbol response of the transmitter output shall be upper-bounded by the temporal mask shown in Figure 6-21. The response shall be measured across a 100-ohm load between tip and ring at the transmitter's W1 interface.

Output before $t = 0$ and after $t = 5.0$ µs shall be <0.032% of the peak amplitude.

In Figures 6-20 and 6-21, the time $t = 0$ is arbitrary.

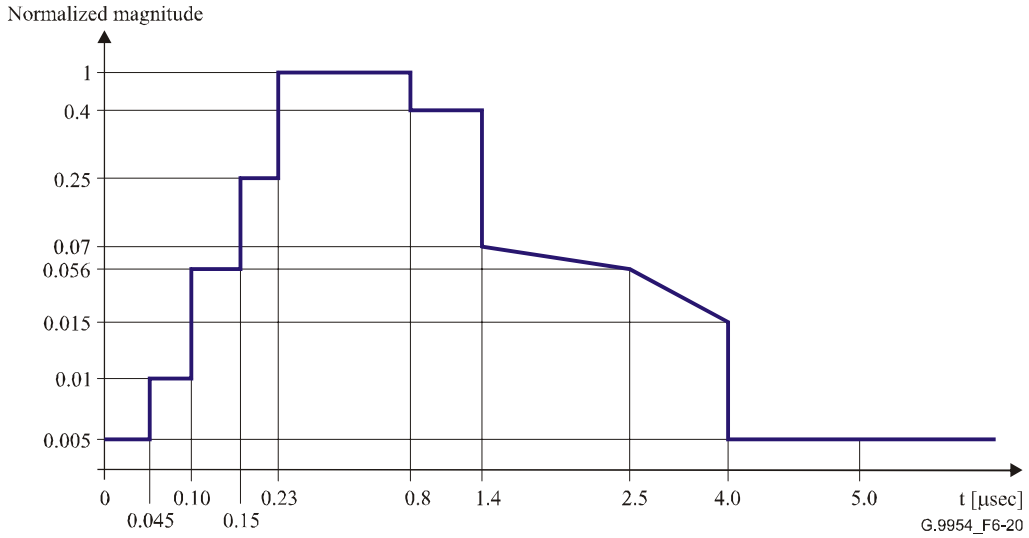


**Figure 6-20/G.9954 – Transmitter symbol response magnitude mask for Spectral Mask #1**

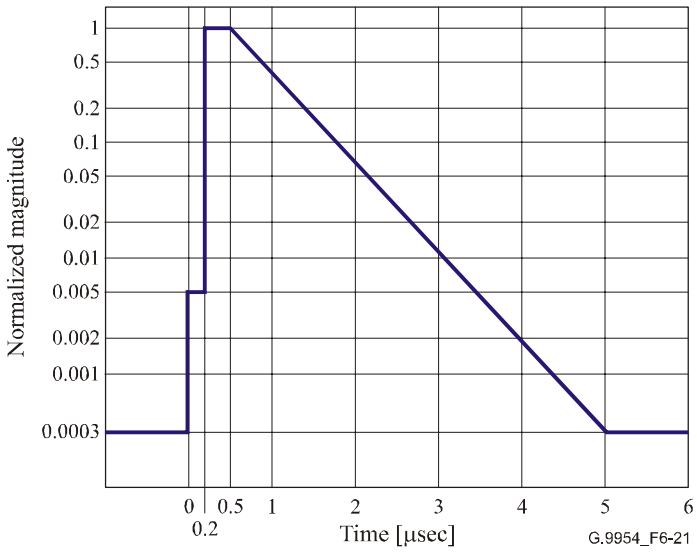


**Figure 6-21/G.9954 – Transmitter symbol response magnitude mask for Spectral Masks #2 and #3**

### 6.8.5 Spurious voice band output

The transmitter C-weighted output in the band extending from 200 Hz to 3000 Hz shall never exceed 10 dBrnC when terminated with a 600-ohm resistive load.

### 6.8.6 Common mode emissions

#### 6.8.6.1 Common-mode output voltage

The transmitter shall emit no more than –55 dBVrms across a 50-ohm load between the center tap of a balun with CMRR > 60 dB and the transceiver ground in the band extending from 0.1 MHz to 50 MHz.

### 6.8.7 Clock tolerance

The transmitter clock frequency shall be accurate to within ±100 ppm over all operating temperatures for the device. The minimum operating temperature range for this requirement shall be 0 to 70 degrees C.

In general, a ±50 ppm crystal will be required to meet this requirement

### 6.8.8 Clock jitter

The RMS jitter of the transmitter clock shall be less than 70 ps, averaged over a sliding 10-microsecond window.

### 6.8.9 I/Q balance

There shall be no gain or phase imbalance in the transmitter, except as noted in 6.5.3.

## 6.9 Receiver electrical specification

### 6.9.1 Receiver sensitivity

#### 6.9.1.1 Maximum signal

The receiver shall detect frames with peak voltage up to –6 dBV across tip and ring at a frame error rate of no greater than $10^{-3}$ with additive white Gaussian noise at a PSD of less than –140 dBm/Hz, measured at the receiver.

#### 6.9.1.2 Minimum sensitivity

The receiver shall detect 1518-octet frames encoded as 2 bits/symbol and 2 Mbaud with RMS voltage as low as 2.5 mV at no greater than $10^{-3}$ frame error rate. The RMS voltage is computed only over time during which the transmitter is active.

The receiver shall detect no more than 1 in $10^4$ 1518-octet, 2 bits/symbol, 2 Msymbol/s frames with RMS voltage less than 1.0 mV.

Both criteria assume additive white Gaussian noise at a PSD of less than –140 dBm/Hz, measured at the receiver, and assume a flat channel.

### 6.9.2 Clock tolerance

The receiver shall meet the requirements of 6.9.4.1 and 6.9.4.2 on loop 1 when the transmitter clock frequency is within ±100 ppm of its nominal value.

### 6.9.3 Immunity to narrow-band interference

#### 6.9.3.1 Differential input

The receiver shall demodulate frames with payload encoded at Spectral Mask #2, 4 MBaud, 3 bits/symbol, and differential RMS voltage as low as 20 mV (measured over the header) at a frame error rate less than $10^{-4}$ under the following conditions:

1) White Gaussian noise with PSD less than –130 dBm/Hz shall be added at the receiver.
2) A single tone interferer with any of the frequency band and input voltage combinations in Table 6-16:

**Table 6-16/G.9954 – Interferer amplitudes**

| Frequency range [MHz] | Maximum peak-to-peak interferer level [volt] |
|---|---|
| 0.01 to 0.1 | 6.0 |
| 0.1 to 0.6 | 3.3 |
| 0.6 to 1.7 | 1.0 |
| 1.7 to 4.0 | 0.1 |
| 7.0 to 7.3 | 0.1 |
| 10.0 to 10.15 | 0.1 |
| 14.0 to 14.35 | 0.1 |
| 18.068 to 18.168 | 0.1 |
| 21.0 to 21.45 | 0.1 |
| 24.89 to 24.99 | 0.1 |
| 28.0 to 29.7 | 0.1 |

The applied voltage shall be measured across tip and ring at the input to the transceiver.

#### 6.9.3.2 Common-mode input

The receiver shall demodulate frames with payload encoded at Spectral Mask #2, 4 MBaud, 3 bits/symbol, and differential RMS voltage as low as 20 mV (measured over the header) at a frame error rate less than $10^{-4}$ under the following conditions:

1) White Gaussian noise with PSD less than –130 dBm/Hz shall be added at the receiver, differential mode.

2) A single-tone interferer, measured between the center tap of a test transformer and ground at the input to the transceiver, with any of the frequency band and input voltage combinations in Table 6-17:

**Table 6-17/G.9954 – Common mode input requirements**

| Frequency range [MHz] | Maximum peak-to-peak interferer level [volt] |
|---|---|
| 0.01 to 0.1 | 20.0 |
| 0.1 to 0.6 | 20.0 |
| 0.6 to 1.7 | 10.0 |
| 1.7 to 4.0 | 2.5 |
| 7.0 to 7.3 | 2.5 |
| 10.0 to 10.15 | 2.5 |
| 14.0 to 14.35 | 2.5 |
| 18.068 to 18.168 | 2.5 |
| 21.0 to 21.45 | 2.5 |
| 24.89 to 24.99 | 2.5 |
| 28.0 to 29.7 | 2.5 |

The common mode rejection of the test transformer used to insert the signal should exceed 60 dB to 100 MHz.

### 6.9.4 System margin requirements

Test loops provided in B.2 shall be used to verify the minimum receiver requirements. The following impairments shall be applied in each loop test: additional (flat) attenuation, additive white Gaussian noise, narrow-band interferers, and 120-Hz impulse noise ("light dimmer noise").

The impairment level (defined in each subclause) must exceed the specified level at each specified payload encoding at the Frame Error Rate (FER) point: $10^{-2}$. A system margin requirement for a single time-varying channel is also defined.

Any entry of "–" in a table implies that there is no requirement under the specified conditions.

#### 6.9.4.1 Attenuation requirements

The attenuator setting described in Table 6-18 is the additional attenuation applied in series with the specified wire loop.

**Table 6-18/G.9954 – Attenuation requirements**

| Loop number | | 1 | 4 | 5 | 6 | 8 | 9 |
|---|---|---|---|---|---|---|---|
| **Payload encoding** | **FER** | **Required impairment attenuator setting [dB]** | | | | | |
| Mask #1, 2 Mbaud, 2 bit/symb | $10^{-2}$ | 34 | 16 | 22 | 11 | 12 | 18 |
| Mask #1, 2 Mbaud, 6 bit/symb | $10^{-2}$ | 30 | 9 | 18 | 6 | 8 | – |
| Mask #2, 4 Mbaud, 3 bit/symb | $10^{-2}$ | 30 | 12 | 17 | 7 | 10 | 16 |
| Mask #2, 16 Mbaud, 3 bit/symb | $10^{-2}$ | 28 | 12 | 13 | – | 8 | – |

### 6.9.4.2 Additive white noise requirements

White noise power at 0 dB attenuator setting: –70 dBm/Hz. The output of the noise attenuator shall be added at the receiver. For loop 1, 20 dB of flat-channel attenuation shall be placed in series with the loop.

**Table 6-19/G.9954 – Additive white noise requirements**

| Loop number | | 1 | 4 | 5 | 6 | 8 | 9 |
|---|---|---|---|---|---|---|---|
| **Payload encoding** | **FER** | **Required impairment attenuator setting [dB]** | | | | | |
| Mask #1, 2 Mbaud, 2 bit/symb | $10^{-2}$ | 42 | 40 | 36 | 46 | 43 | 39 |
| Mask #1, 2 Mbaud, 6 bit/symb | $10^{-2}$ | 58 | 57 | 53 | 63 | 60 | – |
| Mask #2, 4 Mbaud, 3 bit/symb | $10^{-2}$ | 48 | 42 | 42 | 52 | 45 | 52 |
| Mask #2, 16 Mbaud, 3 bit/symb | $10^{-2}$ | 57 | 51 | 52 | 65 | 56 | – |

### 6.9.4.3 Narrow-band interference requirements

Narrow-band interference peak-to-peak amplitude at 0 dB attenuator setting: 2.0 volts at 7.0, 7.3, 10.1, 14.0, 14.35, 18.1, 21.0 MHz. White Gaussian noise is simultaneously applied at a level of –135 dBm/Hz.

**Table 6-20/G.9954 – Narrow-band interference requirements**

| Loop number | | 1 | 4 | 5 | 6 | 8 | 9 |
|---|---|---|---|---|---|---|---|
| **Payload encoding** | **FER** | **Required impairment attenuator setting [dB]** | | | | | |
| Mask #1, 2 Mbaud, 2 bit/symb | $10^{-2}$ | 26 | 26 | 26 | 26 | 26 | 26 |
| Mask #1, 2 Mbaud, 6 bit/symb | $10^{-2}$ | 26 | 30 | 26 | 32 | 30 | – |
| Mask #2, 4 Mbaud, 3 bit/symb | $10^{-2}$ | 26 | 26 | 26 | 26 | 26 | 28 |
| Mask #2, 16 Mbaud, 3 bit/symb | $10^{-2}$ | 26 | 26 | 26 | 43 | 31 | – |

### 6.9.4.4 Impulse noise requirements

Impulse noise peak-to-peak amplitude at 0 dB attenuator setting: 3.0 volts. White Gaussian noise is simultaneously applied at a level of –135 dBm/Hz. The impulse shall be defined as two cycles of a 5.0-MHz square wave summed with four cycles of a 7.0-MHz square wave.

**Table 6-21/G.9954 – Impulse noise requirements**

| Loop number | | 2 | 9 |
|---|---|---|---|
| **Payload encoding** | **FER** | **Required impairment attenuator setting [dB]** | |
| Mask #1, 2 Mbaud, 2 bit/symb | $10^{-2}$ | 3 | 3 |
| Mask #1, 2 Mbaud, 6 bit/symb | $10^{-2}$ | 3 | – |
| Mask #2, 4 Mbaud, 3 bit/symb | $10^{-2}$ | 3 | 3 |
| Mask #2, 16 Mbaud, 3 bit/symb | $10^{-2}$ | 3 | – |

#### 6.9.4.5 Dynamic channel system margin requirement

The receiver shall detect no more than five 1518-octet frames in error out 3000 when sent at a rate of 5 frames per 10 ms over loop #2 under the following conditions:

– During this test, the 330 pF capacitor terminating one of the stubs shall be switched in and out of the loop once per second, i.e., an open-circuit termination shall be used for a period of 1 second every 2 seconds.

– White noise at a level of –140 dBm/Hz shall be added at the receiver.

– The PE shall be Spectral Mask #2, 16 MBaud, 3 bits/symbol.

Switching a capacitor in and out of the loop simulates a switch-hook transition on a common telephone.

#### 6.9.4.6 Telephony ringing signal performance

The PNT device shall be able to accommodate a telephony ringing signal event from a telephone central office. The signal shall consist of a 20-Hz sinusoid with a level of 90 Vrms superimposed on a DC bias level of –52 V (min.). The device shall be immune from a telephony ringing signal that is continuously repeated with an on-time of 2 seconds and an off-time of 4 seconds through the circuit defined in Figure 6-22.

The signal is injected into the circuit through two 500-ohm resistors as shown in Figure 6-22. Since most attenuators have low impedance at DC and could significantly reduce the ringing voltage, two 0.01 µF capacitors are required to provide DC isolation.

When subjected to the telephony ringing signal as defined above, the device frame error rate shall not exceed 0.1% when measured over 100 000 maximum-MTU UDP frames at Spectral Mask #2, 4 MBaud, 3 bits/symbol.

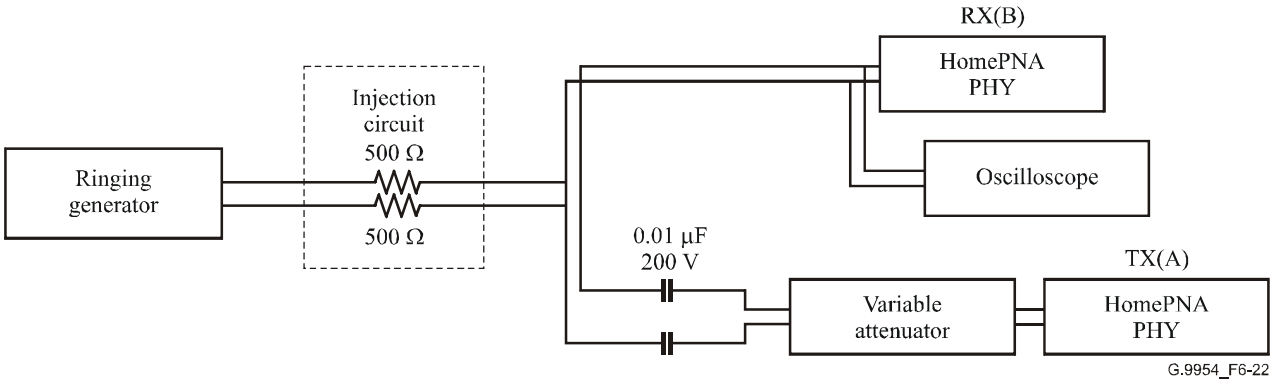


**Figure 6-22/G.9954 – Telephony ringing signal conditions**

### 6.10 Input impedance

#### 6.10.1 Passband return loss

Stations shall conform to the impedance mask corresponding to the highest spectral mask they can transmit.

For stations capable of transmitting with Spectral Mask #2, the average return loss of the transceiver with respect to a 100-ohm resistive load shall exceed 12 dB between 4.75 and 20.25 MHz. This requirement applies to the transceiver powered on or in low-power mode (transmitter powered off). The average return loss with respect to a 100-ohm resistive load shall exceed 6 dB between 4.75 and 20.25 MHz with the transceiver removed from a source of power.

For stations capable of transmitting with Spectral Mask #3, the average return loss of the transceiver with respect to a 100-ohm resistive load shall exceed 12 dB between 4.75 and 27.25 MHz. This requirement applies to the transceiver powered on or in low-power mode (transmitter powered off). The average return loss with respect to a 100-ohm resistive load shall exceed 6 dB between 4.75 and 27.25 MHz with the transceiver removed from a source of power.

### 6.10.2 Stopband input impedance

Stations shall conform to the impedance mask corresponding to the highest spectral mask they can transmit.

Stations capable of transmitting with Spectral Mask #2 only shall have input impedance magnitude greater than 10 ohms from 0 to 30 MHz and shall conform to the lower-bound mask in Table 6-22.

**Table 6-22/G.9954 – Input impedance lower bound mask for Spectral Mask #2**

| Frequency range [kHz] | Minimum impedance [ohms] |
|---|---|
| $0 < f \leq 0.285$ | 1 M |
| $0.285 < f \leq 2.85$ | 100 k |
| $2.85 < f \leq 28.5$ | 10 k |
| $28.5 < f \leq 95$ | 4.0 k |
| $95 < f \leq 190$ | 2.0 k |
| $190 < f \leq 285$ | 1.4 k |
| $285 < f \leq 380$ | 1.0 k |
| $380 < f \leq 475$ | 850 |
| $475 < f \leq 570$ | 700 |
| $570 < f \leq 665$ | 600 |
| $665 < f \leq 760$ | 525 |
| $760 < f \leq 855$ | 450 |
| $855 < f \leq 950$ | 400 |
| $950 < f \leq 1000$ | 350 |
| $1000 < f \leq 1400$ | 175 |
| $1400 < f \leq 2300$ | 100 |
| $2300 < f \leq 2850$ | 50 |
| $2850 < f \leq 3085$ | 25 |
| $3085 < f \leq 4000$ | 10 |
| $4000 < f \leq 4750$ | 30 |
| $20\,250 < f \leq 21\,000$ | 30 |
| $21\,000 < f \leq 25\,000$ | 25 |
| $25\,000 < f \leq 30\,000$ | 50 |

Stations capable of transmitting with Spectral Mask #3 shall have input impedance magnitude greater than 10 ohms from 0 to 30 MHz and shall conform to the lower-bound mask in Table 6-23.

**Table 6-23/G.9954 – Input impedance lower bound mask for Spectral Mask #3**

| Frequency range [kHz] | Minimum impedance [ohms] |
|---|---|
| $0 < f \leq 0.285$ | 1 M |
| $0.285 < f \leq 2.85$ | 100 k |
| $2.85 < f \leq 28.5$ | 10 k |
| $28.5 < f \leq 95$ | 4.0 k |
| $95 < f \leq 190$ | 2.0 k |
| $190 < f \leq 285$ | 1.4 k |
| $285 < f \leq 380$ | 1.0 k |
| $380 < f \leq 475$ | 850 |
| $475 < f \leq 570$ | 700 |
| $570 < f \leq 665$ | 600 |
| $665 < f \leq 760$ | 525 |
| $760 < f \leq 855$ | 450 |
| $855 < f \leq 950$ | 400 |
| $950 < f \leq 1000$ | 350 |
| $1000 < f \leq 1400$ | 175 |
| $1400 < f \leq 2300$ | 100 |
| $2300 < f \leq 2850$ | 50 |
| $2850 < f \leq 3085$ | 25 |
| $3085 < f \leq 4000$ | 10 |
| $4000 < f \leq 4750$ | 30 |
| $27\,250 < f \leq 28\,000$ | 30 |
| $28\,000 < f \leq 32\,000$ | 25 |
| $32\,000 < f$ | 50 |

This requirement applies to the transceiver powered on, in low-power mode (transmitter powered off), or removed from a source of power.

## 7 Media Access Protocol Specification

The G.9951/2 specification (as described in reference [1]) describes a Media Access Control (MAC) protocol that is asynchronous, priority-based and uses CSMA/CD and collision resolution techniques to arbitrate access to the media and resolve media collisions. G.9951/2's support for priority-based media access provides a basic Quality of Service (QoS) mechanism that allows services to be relatively ordered according to priority. To provide QoS guarantees with strictly bounded latency and jitter characteristics, such as those required by voice, streaming audio and video services, a MAC protocol is required that can guarantee media access timing and eliminate the occurrence of events, such as media collisions, that can affect performance guarantees.

The G.9954 MAC protocol is a synchronous protocol that includes an asynchronous MAC mode of operation that is compatible with (and based on) the G.9951/2 Asynchronous MAC protocol. Media access in a network containing a master device is coordinated under the master's control using a Media Access Plan (MAP). The Media Access Plan is broadcast periodically by the master as a Link Layer protocol message (the MAP message). The MAP is used to divide media access time into a sequence of transmission opportunities (TXOPs) whose start times are precisely timed and are of a length sufficient for meeting QoS demands of different services. TXOPs may be allocated to a specific service (or service group) or network node or group of nodes. Using this method, G.9954 nodes are able to avoid collisions by guaranteeing that they will never transmit during media time allocated specifically to another node and by constraining their own transmissions within the limits of the TXOP allocated to them.

This clause describes the G.9954 MAC protocol, including the Asynchronous and Synchronous MAC functions used to coordinate access to the shared media. Switching between Synchronous and Asynchronous MAC modes is also described here.

## 7.1 Modes of operation

The G.9954 MAC shall support two modes of operation:

1) Synchronous MAC (SMAC) Mode – used in a network containing a device acting in the role of G.9954 network master;

2) Asynchronous MAC (AMAC) Mode – used in a network where there is no acting G.9954 network master.

In the presence of a G.9954 master on the network, all G.9954 devices shall operate in SMAC mode. Otherwise they shall operate in AMAC mode.

The presence of a G.9954 master on the network is detected by the reception of Media Access Plan (MAP) messages. Upon reception and decoding of a MAP message, a G.9954 device shall cease operation in AMAC mode and shall continue operation in SMAC mode. The mode switch shall occur within MAC_MODE_SWITCH_TIMELIMIT time-units after receiving the MAP message.

This implies that once a MAP message is received and decoded, subsequent media access shall only be performed in accordance with timing described in the advertised MAP.

MAP messages are expected periodically when operating in SMAC mode. Failure to receive a MAP message within SYNC_TIMEOUT (see 7.3.8) interval from the last MAP message shall be detected by a G.9954 node as master termination. A master may also signal its intention to terminate its role as network master to other devices on the network using a signalling bit (SMAC_EXIT) in the MAP message (see 7.3.3.3). Upon detection of master termination or intention to terminate operation, a G.9954 node shall exit SMAC mode and continue operation in AMAC mode.

The G.9954 MAC modes of operation and the transitions between modes are described in the following state-transition diagram (Figure 7-1).

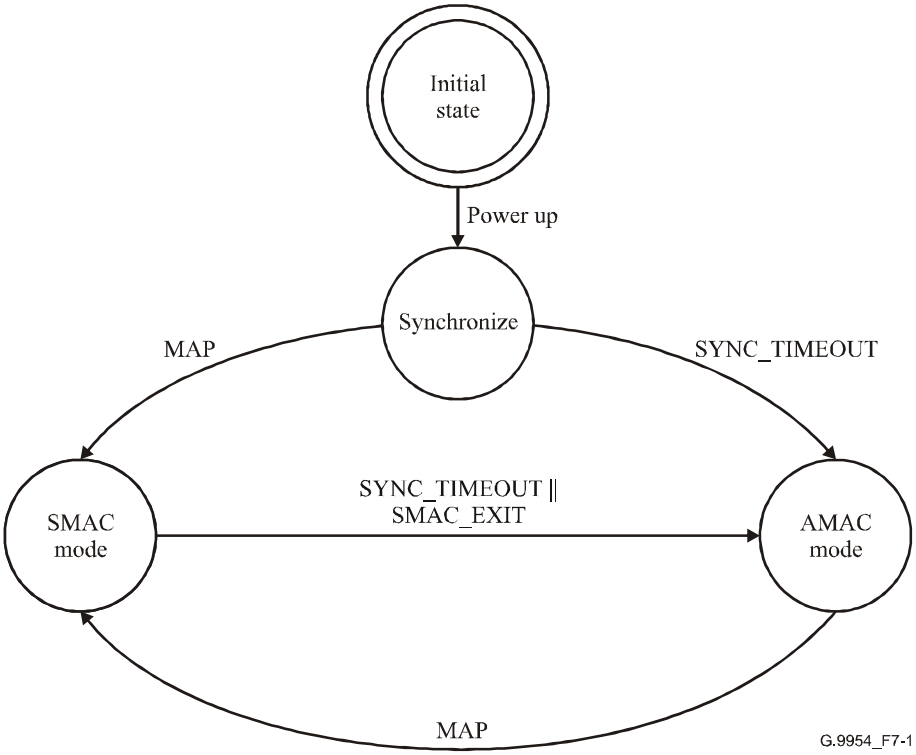


**Figure 7-1/G.9954 – G.9954 modes of operation – State diagram**

When a G.9954 device powers up, it shall first attempt to synchronize with an existing MAC cycle by waiting for a MAP message for up to SYNC_TIMEOUT interval. If a MAP message arrives within SYNC_TIMEOUT interval, a master-controlled network is assumed and the G.9954 device shall enter into SMAC mode. If a MAP message is NOT received within SYNC_TIMEOUT interval, a master-less network is assumed and the G.9954 device shall enter into AMAC mode.

NOTE – This implies that after power-up, a G.9954 device shall NOT start transmitting in AMAC mode until at least SYNC_TIMEOUT interval has elapsed.

The current mode of operation on a G.9954 device is indicated by a flag in the G.9954 Link Layer Capability and Status Announcement (CSA) message. See the Capability and Status Announcement flags in the G.9954 Link Layer Specification in clause 10.

If the network is detected as being master-less, a G.9954 device that is capable of acting in the role of master (a master-capable device), shall attempt to assume the role of network master by signalling its intention using the Link Layer master SELECTION protocol.

For further information on the master Selection Protocol, see the G.9954 Link Layer Specification in clause 10.

### 7.1.1 Synchronous MAC (SMAC) mode

When operating in SMAC mode, a G.9954 device shall ONLY perform media access within dedicated transmission opportunities (TXOPs) described in the MAP. A G.9954 device may transmit within a TXOP if the TXOP is allocated to the device or to a group that the device belongs to. All devices may transmit within spare (UNALLOCATED) TXOPs on a contention basis.

For more information on TXOPs, TXOP assignment and device groups, see 7.3.3.4 and its subclauses.

### 7.1.2 Asynchronous MAC (AMAC) mode

When operating in AMAC mode, a G.9954 device performs media access the same as specified for the G.9951/2 MAC protocol albeit at potentially higher payload bauds and using packet aggregation (frame bursting) to improve protocol efficiency.

For a full specification of G.9954 AMAC mode, see 7.2. For a specification of Packet Aggregation, see 7.4.

### 7.1.3 Switching between Synchronous and Asynchronous MAC modes

A G.9954 device shall switch between SMAC and AMAC modes and vice versa in response to the appearance or disappearance of a G.9954 master on the network. Switching is transparent in the sense that it does not affect a G.9954 device's ability to transmit or receive data although it may affect media access timing and, as a consequence, Quality of Service and network throughput.

When switching from SMAC to AMAC modes, a G.9954 device shall continue to transmit data associated with a service according to AMAC protocol rules using the AMAC priority mechanism. When switching from AMAC to SMAC modes, a device shall set up flows for services requiring QoS guarantees and shall transmit data according to SMAC protocol rules within dedicated TXOPs.

Switching between SMAC and AMAC modes shall not cause the termination of a service unless requested by upper protocol layers.

## 7.2 Asynchronous MAC mode operation

Each station on a PNT network segment, when not in Synchronous MAC mode, executes the Asynchronous MAC function to coordinate access to the shared media. Switching between Synchronous mode and Asynchronous mode is described in 7.1.3.

The MAC timing parameters for Asynchronous Mode are defined in Table 7-1.

**Table 7-1/G.9954 – MAC parameter**

| Clause | Parameter | Min | Max | Units |
|---|---|---|---|---|
| 7.2.1 Basic CSMA | NOMINAL_RMS_VOLTAGE | 100 | – | mVrms |
| | CS_RANGE | 38 | – | dB |
| | CS_IFG | 29.0 – Δ | 29.0 + Δ | microseconds |
| | CS_DEFER | – | 12.0 | microseconds |
| | minFrameSize | 64 | – | octets |
| | maxFrameSize | 1526 | See 7.2.7.1 | octets |
| | TX_FRAME | 92.5 | See 7.2.7.1 | microseconds |
| | Round-trip time (RTT) for 1000 ft | | 3.0 | microseconds |
| | TX_ON | 0 | 4.0 | microseconds |
| 7.2.2 Priority access | PRI_SLOT | 21.0 – Δ | 21.0 + Δ | microseconds |

**Table 7-1/G.9954 – MAC parameter**

| Clause | Parameter | Min | Max | Units |
|---|---|---|---|---|
| 7.2.4 Collision detection | CD_FRAG | 70.0 – Δ | 70.0 + Δ | microseconds |
| | CD_MIN | 32.0 | – | microseconds |
| | CD_THRESHOLD | – | 92.0 | microseconds |
| | CD_RANGE | 36 | – | dB |
| | CD_OFFSET_EARLY | – | 12.0 | microseconds |
| | CD_OFFSET_LATE | – | 15.0 | microseconds |
| 7.2.5 Collision resolution during AMAC mode | attemptLimit | 256 | 256 | |
| | SIG_SLOT | 32.0 – Δ | 32.0 + Δ | microseconds |

The CSMA/CD media access method is the means by which two or more stations share a common transmission channel. To transmit, a station waits (defers) for a quiet period on the channel (that is, no other station is transmitting) and then sends the intended message modulated as per the PHY specification. The transmission deferral is ordered by up to eight priority levels, implementing absolute priority among stations contending for access. If, after initiating a transmission, the message collides with that of another station, then each transmitting station ceases transmission and resolves the collision by choosing a Backoff Level and defers to other stations that have chosen a lower Backoff Level. The distributed algorithm for choosing Backoff Level guarantees that the access latency is tightly bounded. Each aspect of this access method process is specified in detail in subsequent clauses of this Recommendation.

See Figure 7-2 for a block diagram of MAC functions in a station. The Carrier Sense block detects the starting and ending times of a valid frame transmission on the wire. This is used to determine when frames are present on the channel, as well as to determine the presence of a BACKOFF20 signal in a Signal Slot. The Collision Detection block detects the presence of a valid frame transmission from some other station during an active transmission, and for all stations, including non-transmitting stations, detects the received fragment that represents a transmission truncated by a collision. The Collision Resolution block implements the distributed algorithm that controls backoff.

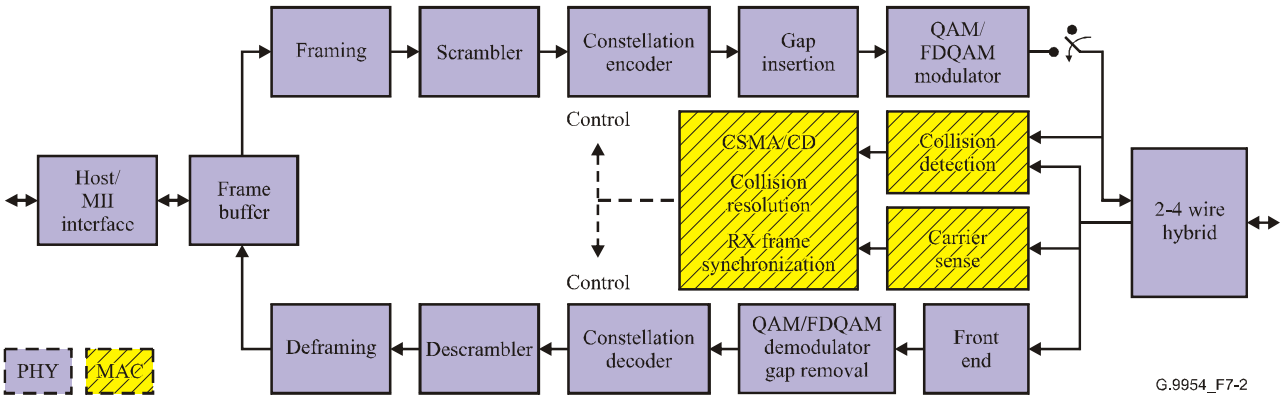


**Figure 7-2/G.9954 – Transceiver block diagram with MAC functions**

Although the performance of the blocks in the MAC function are implementation dependent, certain minimum performance requirements are specified in the following clauses to ensure interoperability and compatible sharing of the channel.

### 7.2.1 Basic CSMA

Basic CSMA behaviour is specified for a transmitter and a receiver.

#### 7.2.1.1 Transmitter behaviour

See Figure 7-3 for a description of a frame transmission that is valid with respect to the specified Carrier Sense (CS) function (Valid CS Frame).

NOTE – A transmitted Valid CS Frame will be affected by various signal impairments when seen by any receiver, and the performance limits of the CS function are implementation dependent.

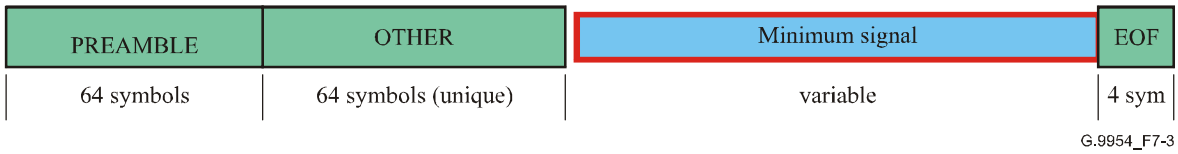


**Figure 7-3/G.9954 – Valid CS Frame**

A Valid CS Frame at the *transmitter* W1 interface has a length of TX_FRAME and consists of:

1) a sequence of symbols whose duration is equal to or greater than 92.5 microseconds (TX_FRAME minimum) duration, but less than the maximum specified in 7.2.7.1;
2) the first (64 + 16 + 24 + 24 + 8) symbols of which modulated at the Base Rate (2-MBaud QPSK, 2 bits per symbol), where the initial 64 symbols consist of the preamble sequence, where the next 64-symbol sequence is unique to the transmitting station, and where the next 8 symbols are the (likely non-unique) bits of the Ethertype field;
3) an arbitrary Minimum Signal, defined as a sequence of symbols whose RMS value over any 8-microsecond window shall never be more than 9 dB less than 100 mVrms across 100 ohms (NOMINAL_RMS_VOLTAGE);
4) 4 symbols of the EOF sequence;
5) a trailing transient, whose peak voltage does not exceed 0.1% of the absolute peak transmitted voltage across a 100-ohm load at the W1 interface at any point >5 microseconds after the last transmitted symbol of the EOF;
6) a gap before the next transmission of this station of CS_IFG microseconds from the last symbol of the EOF to the first symbol of PREAMBLE of the next transmission, measured at the transmitter's W1 interface.

When a station detects what may be a collision, it shall terminate transmission early (see 7.2.4).

A Valid Collision Fragment at the *transmitter* W1 interface consists of:

1) a sequence of symbols of 70.0 µs (CD_FRAG) duration;
2) (64 + 16 + 24 + 24 + 8) symbols modulated at the Base Rate (2-MBaud QPSK, 2 bits per symbol), where the initial 64 symbols consist of the preamble sequence, and where the next 64 symbol sequence is unique to the transmitting station, followed by 8 more symbols;
3) 4 symbols of the EOF sequence;
4) a trailing transient, whose peak voltage does not exceed 0.1% of the absolute peak transmitted voltage across a 100-ohm load at the W1 interface at any point >5 microseconds after the last transmitted symbol of the EOF;
5) a gap of at least CS_IFG + CD_THRESHOLD microseconds from the first symbol of the PREAMBLE64 of the Valid Collision Fragment to the first symbol of the BACKOFF20 signal in the first Backoff Signal Slot (if present), measured at the transmitter's W1 interface.

Receivers are only required to correctly detect Valid CS Frames, Valid Collision Fragments, and the BACKOFF20 signal described in 7.2.5.

The Inter-frame Gap shall be 29.0 µs (CS_IFG), where the gap is defined at the points at which the previous frame drops below 50% of its peak and the current frame rises above 50% of its peak.

### 7.2.1.2 Receiver behaviour

Timing of subsequent transmissions following a Valid CS Frame or Valid Collision Fragment are based on a MAC timing reference, established by the receiver. Time following a transmission is divided into *slots*: an Interframe Gap (IFG); three Backoff Signal Slots (following collisions); and 8 priority slots (see Figures 7-4 and 7-5). During these time periods the MAC is *synchronized* and the slot timing is defined by the rules for valid transmissions in the previous clause. After priority slot 0, there may be an arbitrarily long period with no transmissions followed by one or more stations attempting transmission. In this latter case the MAC is *unsynchronized*.

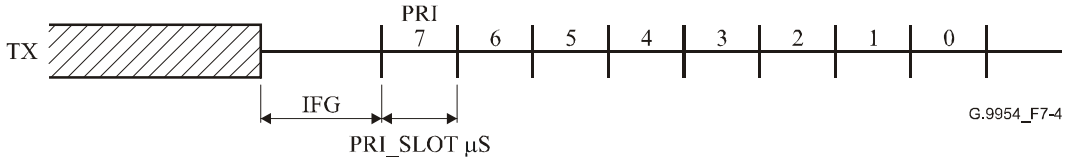


**Figure 7-4/G.9954 – Priority slots**

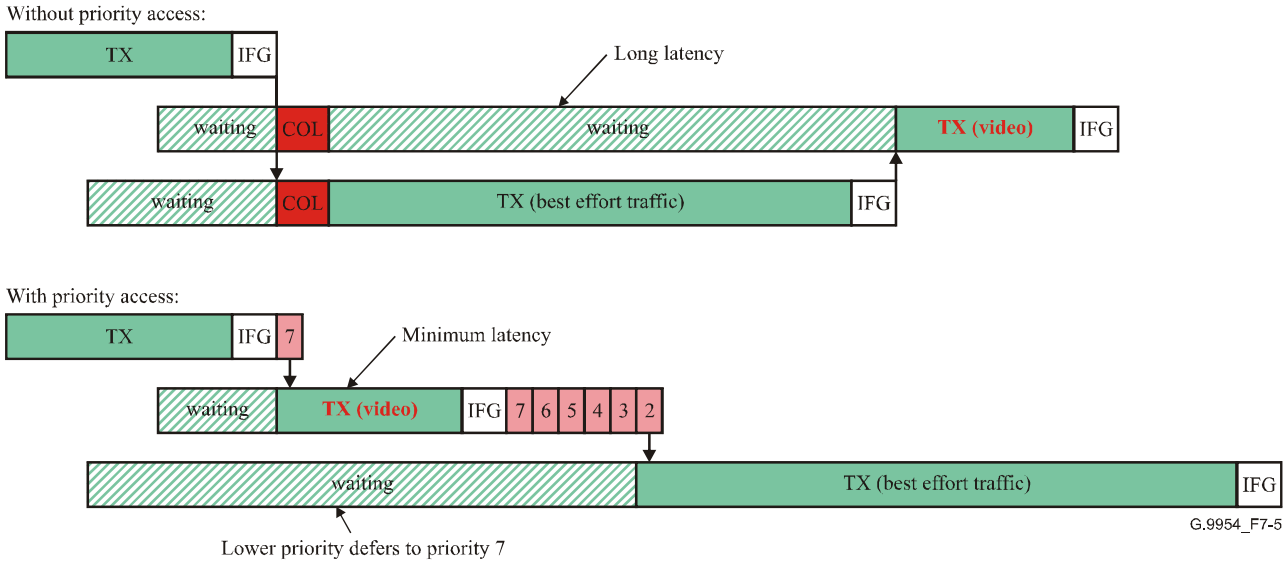


**Figure 7-5/G.9954 – Example of priority access**

When MAC timing is synchronized, stations shall commence any transmission no earlier than 0 and no later than 4 µs (TX_ON) after a slot origin, measured at the transmitter W1 interface.

The receiver Carrier Sense function shall detect a maximum-amplitude Valid CS Frame over a range of 0 to at least 38 dB (CS_RANGE) flat-channel insertion loss and additive noise with a flat PSD of –140 dBm/Hz at the receiver with a missed frame rate of less than $10^{-4}$ and a premature end-of-frame declaration rate less than $10^{-4}$; see 6.9.1. With additive white Gaussian noise applied at the input with a PSD of –110 dBm/Hz, the false carrier detection rate shall be no greater than 1 per second.

When the MAC is unsynchronized, the latest a station may commence transmission after a possible Valid CS Frame has appeared at the W1 interface shall be 12 µs (CS_DEFER) from the first symbol

of the PREAMBLE64 of the detected frame, as measured at the station's W1 interface. CS_DEFER is the maximum allowed carrier sense delay.

### 7.2.2 Priority access

The PNT system can be used for carrying media streams, such as audio and video. To reduce the latency variation in these streams, a priority mechanism is implemented to allow higher layers to label outgoing frames with priority, and guarantee that those frames will have preferential access to the channel over lower priority frames. The access priority method implemented is to delay transmissions to a *slot* beyond the minimum inter-frame gap, based on the priority level of the frame waiting to be transmitted.

Slots are numbered in decreasing priority, starting at priority 7. Higher priority transmissions commence transmission in earlier slots and acquire the channel without contending with the lower priority traffic. A station's *Priority Slot* is based on the PHY priority number associated with the frame ready for transmission (PRI), as determined by the network stack and communicated to the MAC; PRI is a field in the Frame Control field, described in 6.3.3. The station uses any slot with a number less than or equal to PRI, normally the slot numbered exactly PRI, and it may only commit to transmit at the start of a Priority Slot, i.e., if a station is ready to transmit a PRI = 7 frame only after the start of Priority Slot 7, it must wait until the start of Priority Slot 6 to transmit. See Figure 7-4 for the relative timing of priority slots. (After Priority Slot 0 there are no more priority slots, and any station with traffic at any priority level can contend on a first-come, first-served basis. All collisions after Priority Slot 0 are considered to happen at PRI = 0.)

The Priority Slot width is 21.0 μs (PRI_SLOT).

No station shall transmit in a priority slot numbered higher than the priority (PRI) assigned to the frame being transmitted.

Stations not implementing priority shall default PRI to a value of 2 when transmitting.

Stations waiting for transmission shall monitor Carrier Sense (CS) and defer if CS was true prior to the start of the next Priority Slot in which it can transmit, or if beyond Priority Slot 0, the station shall defer if CS was true. Any station ready to transmit at the start of the next Priority Slot in which it can transmit shall transmit at the start of that Priority Slot without deferring if CS was false prior to the start of that Priority Slot.

See Figure 7-5 for an example of video traffic at priority level 7 gaining access ahead of best effort traffic scheduled at level 2.

The slot timer is restarted if there is some other transmission that acquires the channel while a station is waiting at a lower priority.

### 7.2.3 Priority mapping

The PRI value is the priority the MAC uses to schedule transmission and is the value present in the PRI field of the frame header. This value is determined by a higher layer in the network stack and the method of priority labeling is outside this Recommendation. The PRI field is used to transport the priority label from source to destination, to assist the destination in managing the receive queue. The 3-bit priority values referred to are "PHY priorities". PRI = 7 has the highest priority, PRI = 0 has the lowest.

There may be a mapping between PHY priorities and the Link Layer (LL) priority values as delivered to the Link Layer by the Network Layer. This mapping is described in the Link Layer Protocols Specification in clause 10.

In general, the IP network layer or application layer will determine what the policy is used to map traffic onto LL priorities. For instance, IETF Integrated Services (RFC 2815) currently defines priority 0 as the default "best effort" priority, and priority 1 as the penalty "worse than best effort"

priority – and most implementations will map best effort to PHY PRI = 2 and worse-than-best-effort to PHY PRI = 0.

The PHY priority mechanism is strict priority (as opposed to schemes which allocate lower priorities some minimum percentage of network capacity) – higher priority traffic always defers lower priority traffic. Higher priority traffic will be limited by admission control or other Link Layer policy mechanism to prevent over-subscription.

### 7.2.4 Collision detection

Two or more stations may begin transmitting in the same Priority Slot following the IFG period. All stations monitor the channel to detect the colliding transmissions of other stations.

NOTE 1 – The colliding frame(s) will be received over a channel with impairments, and the performance of Collision Detection is implementation dependent, within the bounds of this Recommendation.

See Figure 7-6. Passive stations can detect collisions by observing the length of transmission fragment and the validity of the received PREAMBLE64.

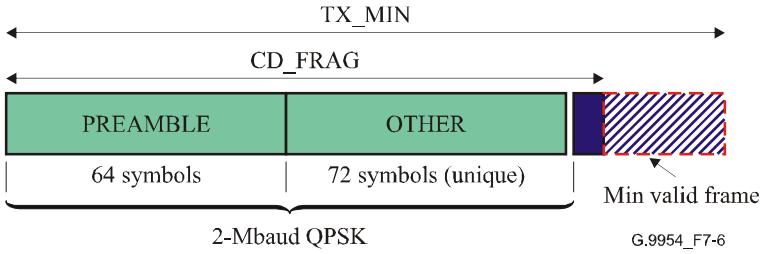


**Figure 7-6/G.9954 – Length of collisions and non-collisions**

A Valid CS Frame is guaranteed to have a unique symbol sequence within the first 128 symbols (which are transmitted at Base Rate). The Ethernet MAC Source Address (SA) is used to guarantee uniqueness. That field is scrambled, but the [scrambled SA, SI] tuple will be unique. SI is the 4-bit scrambler initialization field, defined in the G.9954 PHY specification in clause 6.

After detecting a collision, a station shall continue to transmit through the Ethertype field followed by an EOF sequence (symbol 139) and then cease transmission.

Thus, a station detecting a collision will cease transmission no later than 70.0 µs (CD_FRAG) after the beginning of the frame as measured at the W1 interface. The minimum size of a Valid CS Frame is 92.5 µs (TX_MIN).

No jam signal is transmitted on collisions.

Passive stations that are not transmitting shall monitor the length of Carrier Sense events and generate a Collision Fragment indication to the Collision Resolution function if the duration of carrier is less than 92 µs (CD_THRESHOLD).

Stations shall not recognize carrier events shorter than 32.0 µs (CD_MIN) as collisions.

All transmitting and passive stations shall be capable of detecting the collision of any maximum-amplitude Valid CS Frame transmission received over a range of 0 to 36 dB (CD_RANGE) flat-channel insertion loss and additive noise with a flat PSD of –140 dBm/Hz at the receiver with a missed-collision error rate of less than $10^{-4}$ and a false collision error rate of less than $10^{-3}$, where the origin of the colliding frame is offset relative to the first symbol of the transmitted frame anywhere from earlier by up to 12 µs (CD_OFFSET_EARLY) to later by up to 15 µs (CD_OFFSET_LATE). The same requirements shall be met on loop 9 shown in Annex B.

NOTE 2 – Where there is a missed collision, the probability of detected and undetected errors in the payload data is enhanced, so Collision Detection implementations should be biased towards false collision errors, which are more innocuous.

### 7.2.5 Collision resolution during AMAC mode

A collision occurs when two or more stations are active with ready frames and are contending for access to the channel at approximately the same time; generally, collisions are between frames at the same priority level. A distributed collision resolution (CR) algorithm is run which results in stations becoming ordered into Backoff Levels where only one station is at Backoff Level 0 and can therefore acquire the channel. After the winning station completes its transmission, all stations reduce their Backoff Level by one if it is greater than zero, and the new station(s) at Backoff Level 0 attempt transmission. All stations, even those with no frame to transmit, monitor the activity on the medium. Also, the collision resolution cycle is closed, so that stations that did not collide are not allowed to contend for access to the medium until all stations that collided have transmitted one frame successfully or have forgone the right to transmit their waiting frame. Ultimately, all stations that were contending for access in the initial collision gain access to the wire and the collision resolution cycle is ended. This results in access latency being tightly bounded.

This mechanism differs from Binary Exponential Backoff (BEB) used in other versions of Ethernet in that the Backoff Level does not determine the contention slot chosen by a station – all stations at a given priority always contend in the slot corresponding to the access priority. Instead, stations at non-zero Backoff Levels defer contending until stations that are at zero Backoff Level transmit. The method used is called Distributed Fair Priority Queuing (DFPQ).

Each station maintains eight Backoff Level (BL) counters, one for each priority. The Backoff Level counters are initialized to 0.

The priority level of a collision can be inferred from the priority slot where the collision occurs.

Consider the case where stations are only contending on one priority. After a collision and an IFG, three special *Backoff Signal* slots (S0…S2) are present before the normal sequence of priority contention slots occurs (see Figure 7-7). Signal slots only occur after collisions; they do not follow successful transmissions.

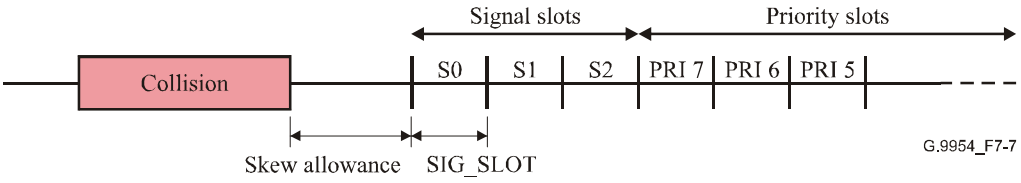


**Figure 7-7/G.9954 – Signal slots**

Each active station pseudo-randomly chooses one of the slots, and transmits a BACKOFF20 signal, defined below; Appendix VII discusses a possible method for generating a pseudo-random slot number. More than one station can transmit a BACKOFF20 signal in the same slot. The active stations transmit BACKOFF20 signals to indicate ordering information that determines the new Backoff Levels to be used.

For transmission on PHY priority 7, a station may optionally use the Collision Management Protocol (defined in 10.12) to dynamically pre-assign a unique sequence of collision slot values that ensures deterministic collision resolution among stations that implement this protocol. For a given PHY priority 7 collision resolution cycle, instead of choosing a slot at random, a station following this protocol would choose successive slots as specified by the pre-assigned Collision Slot Sequence. A value in the range [0,2] for s<x> indicates a specific signalling slot to be used

following the <x>th collision, while the value of 3 indicates the use of a random value chosen by the station at the time of the collision.

All stations (even those without a frame ready to transmit) monitor collision events and the Backoff Signal slots to compute the Backoff Level. If an active station sees a BACKOFF20 signal in a slot prior to the one it chose, it shall increase its Backoff Level. Those stations at Backoff Level 0 (ones that are actively contending) that saw no BACKOFF20 signals prior to the one they chose, shall remain at Backoff Level 0 and contend for transmission in the priority slot equal to PRI that immediately follows the BACKOFF20 signal sequence. Eventually, only one station remains at Backoff Level 0 and successfully gains access to the channel. Stations with higher priority waiting frames may pre-empt the collision resolution by transmitting in a higher-priority slot.

All stations, even those not contending for access to the wire, also maintain a Maximum Backoff Level (MBL) counter per priority, which is incremented for each BACKOFF20 signal seen and decremented when a successful transmission occurs. The MBL is non-zero whenever a collision resolution cycle is in progress. When a station first becomes active, if MBL is non-zero, BL is initialized to contents [MBL], otherwise BL is initialized to 0. This ensures that all currently active stations gain access to the channel before stations can re-enter the waiting queue.

The BACKOFF20 signal shall be a symbol sequence consisting of 16 symbols of the preamble sequence (TRN16) transmitted, followed by the 4-symbol EOF sequence. Stations shall detect the BACKOFF20 signal(s) in a Backoff Signal slot even if more than one station selects the same slot.

Stations shall implement saturating 4-bit BL and MBL counters.

The width of the Signal Slot shall be 32.0 μs (SIG_SLOT). The first Signal Slot is timed from the detected start of the first of the overlapping collision fragments.

Stations shall implement the MAC function with collision resolution whose behaviour matches the procedural model described in the next clause. The timing reference in the pseudo-code in 7.2.6 is referenced to the carrier sense signal, not to the signal at the W1 interface.

### 7.2.6 Media access procedural model

The procedural model uses a pseudo-code modeled after Concurrent Pascal. See IEEE Std 802.3 1998 Clause 4.2.2 for an overview of this pseudo-code. Some additional but obvious liberties have been taken with the syntax used here. The pseudo-code assumes there is no time delay between carrier sensing and the arrival of the signal at the W1 interface. Thus, implementations must account for additional implementation delays.

The code models three independent concurrent processes (Deference, Transmitter, Receiver), which interact through shared variables. The Deference process is driven by the detection of transmissions on the channel, and times the boundaries for Signal Slots and Priority Slots. The shared variable `currentPriority` signals the Transmitter process when a transmission slot exists.

```
{Deference:
 Loop, looking for carrier sense, and when found determine whether the
 transmission was a collision or valid frame.

 If it was a collision, process the signal slots and run the
 collision resolution algorithm.
 In any case, then process the priority slots, looking for carrier.

 The "current" priority level is sticky from the slot the last
 collision occurred in.

 The Backoff Level (BL) and Maximum Backoff Level (MBL) counters
 are saturating at 0 and 15.}

Const
```

```
   nPriorities = 8;   {Number of priority levels}
   nSignals = 3; {Number of signal slots}
   nLevels = 16; {Number of Backoff Levels}

process Deference;
begin
   currentPriority := 0; {Priority of the slot we are in}
   cycle {deference loop}
      sawFrame := false;
      sawCollision := false;
      while not carrierSense() do nothing; {watch for carrier to appear}
      deferring := true;
      startTime := time();
      stopTime := startTime;
      while carrierSense() do
          stopTime := time();
      if ((stopTime - startTime > CD_MIN) and
          (stopTime - startTime < CD_THRESHOLD)) or collisionSense()
          then sawCollision := true
      else sawFrame := true;
      {After a collision, process the three signal slots}
      if sawCollision then
      begin
         {wait until the end of the IFG, timing from start of fragment
            reduces skew, since start-of-carrier uncertainty is less than
            end-of-carrier uncertainty }
         while (time() - startTime < CS_IFG + CD_THRESHOLD) do nothing();
         computeSignals();
         for (i := 0; i < nSignals; i++)
         begin
            startTime := time();
            signal[i] := 0;
            if signalSlot = i then sendSignal();
            while (time()-startTime < SIG_SLOT) do
               if carrierSense() then signal[i] := 1;
         end;
         processSignals();
      end;
      if (not sawCollision) then
      begin
         {wait until the end of the IFG}
         while (time() - stopTime < CS_IFG) do nothing();
         {If last transmission was successful, drop Backoff Levels}
         BL[currentPriority] := saturate(0,nLevels-1,BL[currentPriority]-1);
         MBL[currentPriority] := saturate(0,nlevels-1,MBL[currentPriority]-1);
      end;
          {avoid timing hazard with transmitter, currentPriority must be setup
            before deferring is cleared}
      currentPriority := nPriorities-1;
      deferring := false;
      {Now time out the Priority (contention) slots}
      for (i := nPriorities-1; i>=0; i--)
      begin
         slotTime := time();
         currentPriority := i;
         while (time()-slotTime < PRI_SLOT) do
            if carrierSense() then endcycle;{restart deference loop}
         {if priority slot passed with no contenders, then that priority
          level must be idle, good practice says make sure the backoff
          counters are reset}
         BL[currentPriority] := 0;
         MBL[currentPriority] := 0;
      end;
   end; {cycle}
```

```
end; {Deference}


{computeSignals: Determine which signals to send}
function computeSignals();
begin
   signalSlot := -1; {-1 means no signal to send, initialization}
   if (txReady and (txPriority = currentPriority) and BL[txPriority]=0) then
       if (txPriority = 7 and activeCSSClient) then
         {Optional CSS-assigned collision slot value for PHY priority 7}
         if (slotSequence[Ncollisions] = 3 or Ncollisions > 8) then
            signalSlot = integerRandom(nSignals); {Use random value}
         else
            signalSlot = slotSequence[Ncollisions]; {Use assigned value}
      else
         {Normal random signal slot selection}
          signalSlot = integerRandom(nSignals); {select Backoff Signal slot}
end; {computeSignals}


{processSignals:  Process the received signals, adjusting the Backoff Levels}
function processSignals();
begin
   psignals := 0;
   for (i=0; i < nSignals; i++)
      if signal[i] then psignals++;
   if (txReady and (txPriority = currentPriority)) then
   begin
      backoffLevel := BL[currentPriority];
      if backoffLevel = 0 then
      begin
         tem := 0;
         for (i=0; i < signalSlot; i++)
            if signal[i] then tem++;
         BL[currentPriority] := saturate(0,nLevels-1,tem);
      end;
      if backoffLevel > 0 then
         if psignals > 0 then
            BL[currentPriority] :=
               saturate(0,nLevels-1,backoffLevel + psignals-1);
   end;
   if psignals > 0 then
   begin
      if MBL[currentPriority] = 0 then MBL[currentPriority] := psignals;
      else MBL[currentPriority] = saturate(0,nLevels-1,MBL[currentPriority]
                                                    + psignals-1);
   end;
end; {processSignals}


{Transmitter:  Wait for txReady and txPriority from the link level process.
 send txFinished when frame has been sent.}
process Transmitter;
begin
   cycle
      while (not txReady) do nothing();
      BL[txPriority] := MBL[txPriority];
      Ncollisions = 0;
      while (not (txPriority >= currentPriority and BL[txPriority]=0)
             or deferring)
         do nothing();
      ttime := time();
      xmtDataOn(); {start data transmitting}
      while xmtBusy() and (time() - ttime < CD_FRAG) do
```

```
      begin
         if collisionSense() then
         begin
            xmtDataOff();{turn off, after sending minimum collision fragment}
            Ncollisions++; {timeout on excessive collision limit}
            if Ncollisions = attemptLimit-1 then txFinished();
            endcycle;
         end;
      end;
      while xmtBusy() do nothing();
      txReady := false;
      txFinished();     {signal link level that frame has been transmitted}
   end; { cycle }
end; { Transmitter }

{collisionSense: }
function collisionSense();
begin
  { When transmitting, detect the presence of a second transmission.
    When receiving, detect overlapped transmissions}
end; { collisionSense }

{Receiver: }
process Receiver;
begin
  { Wait for carrier sense.  Demodulate received signals into frames.
    Reject collision fragments. Determine frame boundaries. Check FCS.
    Filter based on destination address. Perform optional Link Layer
    signalling and other controller functions.}
end; { Receiver }
```

### 7.2.7 Asynchronous MAC parameters

This clause is determinative of AMAC parameters, to supersede any other value of these parameters in other parts of this Recommendation. Where a tolerance is indicated, $\Delta$ = 63 nanoseconds (see Table 7-1).

#### 7.2.7.1 Minimum and maximum link-level frame sizes

The link-level frame consists of the DA through FCS fields, prior to the PHY-level frame encapsulation. All PNT stations shall transmit link-level frames with a minimum of 64 octets. The payload field of link-level frames smaller than minFrameSize shall be padded with any value octets appended after the supplied payload to make the frame minFrameSize long.

The maximum standard Ethernet frame is 1518 octets, but some PNT link-layer encapsulations may add additional octets.

All PNT stations shall be able to transmit and receive link-level frames with up to 1526 octets. No PNT station shall transmit link level-frames with more than 512 × bits per symbol × baud octets. The number of octets specified counts DA through FCS, and does not count preamble, header, CRC-16 or PAD or EOF. This will result in a maximum frame duration (maximum TX_FRAME value) of 4166 microseconds for a frame with PE = 15. A G.9954 station shall default the maximum length frame it will send to a given DA to 1526 octets until it can determine that the receiver can support larger transmission units (e.g., by use of the CSA announcement of CSA_MTU, see "Link Protocols for G.9954").

These maximums establish an upper bound on the duration of a given transmission and an upper bound on the maximum frame size that receivers must accommodate.

## 7.3 Synchronous MAC mode operation

Each station on a G.9954 network segment, in the presence of a G.9954 master device, shall execute the Synchronous MAC function to coordinate access to the shared media.

MAC timing parameters for SMAC mode are based on the same timing parameters as for AMAC mode and are defined in 7.3.7 and 7.3.8.

Media access in a synchronous network is controlled by the master using a Media Access Plan (MAP). The MAP specifies media access timing on the network. Media access time, in the MAP, is broken up into transmission opportunities (TXOPs) of a specified length and start time that are allocated to specific network devices in accordance with their resource demands. Media access timing is planned by the master in such a manner so as to normally avoid collisions. Collisions may, however, still occur within transmission opportunities that are designated as contention periods (CPs). Within contention periods, the media access method used by G.9954 nodes shall be based on the AMAC function, except that transmissions are bounded to the end of the contention period. Collisions are similarly resolved using the AMAC collision resolution method (DFPQ), described in 7.2.5. The collision resolution process is either bounded or unbounded within the contention period, depending upon policy decision by the master as signalled in the MAP.

The details of the SMAC protocol are described in the following clauses.

### 7.3.1 Network devices and device identifiers (DEVICE_ID)

G.9954 devices are identified by their globally unique 48-bit universal MAC address.

G.9954 devices that require QoS contracts shall REGISTER with the G.9954 master and identify themselves using their globally unique 48-bit universal MAC address. The MAC address is used by the master, during network admission, as a unique key for device identification.

A G.9954 network device that has been admitted to the G.9954 synchronous network by the master is assigned a *short* address, known as the DEVICE_ID. The DEVICE_ID is used to identify the assignment of TXOPs to devices. A G.9954 device is informed of its assigned DEVICE_ID by the master during the Network Admission Protocol (see 10.15).

NOTE – The short address form is used for both protocol efficiency and in order to guarantee that devices requiring QoS guarantees go through an explicit network admission process in order to utilize media resources.

The network DEVICE_ID is a 6-bit structure with values in the range 0 to 63. DEVICE_IDs are unique within the network.

The following DEVICE_ID are defined:

**Table 7-2/G.9954 – DEVICE_ID definition**

| Device name | DEVICE_ID | Description |
|---|---|---|
| NULL DEVICE | 0 | The NULL (undefined) DEVICE_ID. |
| Master DEVICE | 1 | Identity of the selected G.9954 network master. |
| Reserved | 2-63 | DEVICE_IDs reserved for assignment to admitted QoS-enabled G.9954 devices. |

### 7.3.2 Service flows and flow identifiers (FLOW_ID)

A service flow (or just *flow* for short) is a simplex logical communication channel between a source and destination device. It is service-oriented and is defined by the type of information it transports. A device may support multiple service flows where each service flow is identified by a FLOW_ID.

A FLOW_ID is a 4-bit number in the range 0-15. A flow is uniquely identified in the network by the tuple (Source Address, Destination Address, Flow ID). This implies that there can be up to 15 flows between a source and destination device given that the FLOW_ID with a value of 0 represents the NULL (undefined) FLOW_ID.

### 7.3.3 Synchronous MAC timing

#### 7.3.3.1 The MAC cycle

Media access in SMAC mode is performed within the context of a periodic MAC cycle. Each period of the MAC cycle starts with the transmission of a Media Access Plan (MAP) by the master and ends at the end of the planned media access period described in the MAP. G.9954 network devices shall synchronize with the MAC cycle by detecting the presence of a MAP message and by performing media access according to the media access plan described in the MAP. The MAP describes the allocation of *Transmission Opportunities* or *TXOPs* to devices and/or service flows in the network. TXOPs are described by their start-time, duration and by the device and/or service that may transmit within the TXOP. Timing references within the MAP are relative to the start of the MAC cycle which the MAP describes. The beginning of the first symbol of the PREAMBLE of the received MAP transmission represents time zero.

The MAP shall describe the TXOPs in the MAC cycle immediately following the cycle in which the MAP is received. This means that the MAP message that starts MAC cycle N describes the TXOPs in MAC cycle N + 1. This is illustrated in Figure 7-8.

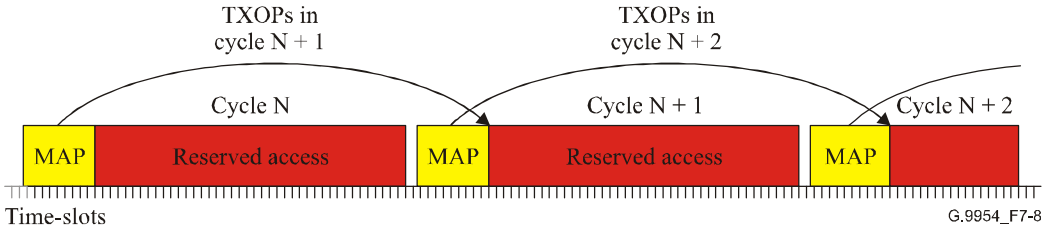


**Figure 7-8/G.9954 – MAC cycle and MAP reference**

MAC cycles are separated by an Inter-Cycle Gap (CS_ICG). An Inter-Cycle Gap is a guaranteed minimum period where the medium is idle based on the carrier sense function. The interval is measured from the last symbol of the EOF of the last frame in a MAC cycle to the first symbol of the PREAMBLE of the MAP transmission. Bursts within a MAC cycle are separated by an Inter-Frame Gap (MAP_IFG) as defined in 7.3.3.3.

The G.9954 master shall allocate media time for the CS_ICG and MAP_IFG and encode it within the definition of the TXOPs described in the MAP. Each TXOP shall contain media time that includes the gap before the next transmission.

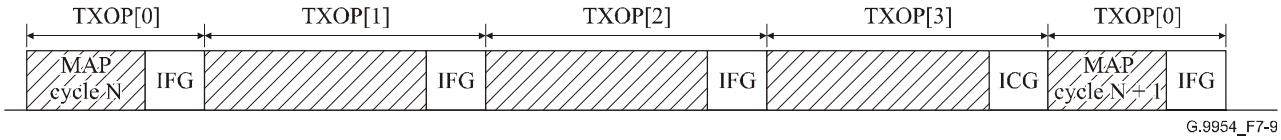


**Figure 7-9/G.9954 – MAP_IFG and CS_ICG accounting**

The actual length of a CS_ICG and MAP_IFG are defined in 7.3.8.

### 7.3.3.2 The MAC cycle length

MAC cycles are periodic and typically of constant length. The actual length of the MAC cycle may vary dynamically between cycles from CYCLE_MIN to CYCLE_MAX depending on scheduling constraints and decisions.

The length of the MAC cycle that a MAP describes is encoded implicitly in the MAP.

Since the MAP describes the Media Access Plan for the next MAC cycle, it always takes two MAC cycles for a MAC cycle length modification to take effect. This is illustrated in Figure 7-10.

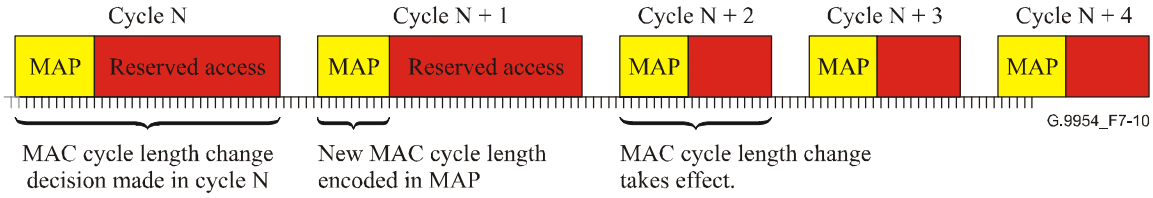


**Figure 7-10/G.9954 – Variable MAC cycle lengths**

### 7.3.3.3 The Media Access Plan (MAP)

The MAP control frame signals the start of a MAC cycle (the "current" MAC cycle) and describes the TXOPs planned in the "next" MAC cycle. The "current" MAC cycle is identified by the Sequence Number contained in the MAP frame starting the cycle. The "next" MAC cycle is the MAC cycle that follows the "current" MAC cycle and contains a Sequence Number that is one more than the "current" MAC cycle accounting for modulo arithmetic.

The extent of the media access plan described by MAP frame is a single MAC cycle only. Since a MAP frame describes the media access plan for the next MAC cycle, a MAP becomes current at the beginning of the next MAC cycle and remains current until to the beginning following MAC cycle. The information in a MAP becomes out of date at the end of the MAC cycle that it describes.

A G.9954 network device shall NOT transmit within a MAC cycle for which it does not hold a valid and current (up-to-date) MAP.

A MAP frame shall be identified by a frame with Frame Type (FT = 0x90) (i.e., Frame Subtype (FS = 0x01)) and has the structure as in Figure 7-11.

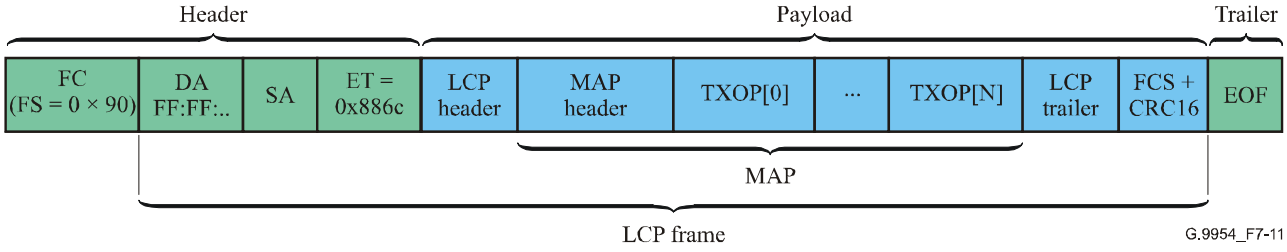


**Figure 7-11/G.9954 – MAP frame structure**

The MAP frame shall be encoded as a Link Layer Control Protocol (LCP) frame with the payload components composed of a fixed length MAP header followed by a variable length table of TXOPs. The number of TXOPs in the MAP is encoded in the MAP header. The size of a MAP Control frame shall not exceed the size of a standard Ethernet frame (i.e., 1500 byte payload).

For a standard Ethernet frame of payload size 1500 bytes, there are 1480 bytes available for the variable length TXOP table (after removing LCP and MAP headers). This means that the number of TXOP entries in the table will not exceed 370. Given a maximum MAC cycle size of 50 ms, and a

minimum frame size of 92.5 µs + 29 µs (GAP), the theoretical maximum number of MAP entries is limited to 50000/(92.5 + 29) = 411 TXOPs. In practice, the number of TXOPs in a MAP is expected to be significantly less than this theoretical limit, on the order of 10s of entries.

NOTE – The Link Control Frame format is used for convenience in order to allow the MAP to be easily passed up to higher protocol layers (possibly residing in the driver stack behind an IEEE 802.3 standard interface).

The MAP frame shall always be sent to the "broadcast" destination address and the entire contents shall be transmitted using the most robust constellation encoding (Mask#2, 2Mbaud and 2 bits per symbol – PE = 33).

The MAP contains the following information in Table 7-3:

**Table 7-3/G.9954 – MAP information**

| Field name | Field size [bits] | Description |
|---|---|---|
| MAP HEADER | 3 × 32 | |
| • ControlField | 32 | Set of control fields used to control the behaviour of endpoint nodes. The encoding of this field is described immediately below: |
| • • Modified | 1 | Indicates that the TXOP table defined in this MAP is different from the TXOP table defined in the "previous" MAP where "previous" is defined as the MAP sent in the "previous" MAC cycle with *Sequence Number* one less than the "current" *Sequence Number (*accounting for modulo arithmetic).<br>0 MAP is the same as "previous" cycle.<br>1 MAP changed since "previous" cycle.<br>This flag may be used by an endpoint for local optimization. |
| • • CycleLatency RepairMethod | 2 | The cycle latency repair method to be used by end-points when the start of the MAC cycle (as indicated by the arrival time of the MAP) is delayed compared to the scheduled arrival time. For further details, see 8.6.4 below.<br>0 None – Don't use latency repair techniques at start of cycle<br>1 Adjust clock – Adjust the clock used to time SMAC transmissions at start of cycle by the delay offset.<br>2-3 Reserved for future use |
| • • Collision Resolution Method | 2 | The collision resolution (CR) method to be used to resolve collisions during TXOPs defined as contention periods.<br>0 DFPQ (G.9951/2-style CR).<br>1 Bounded-DFPQ (DFPQ with CR bounded within the CTXOP or UTXOP).<br>2,3 Reserved for future use.<br>For further details on collision resolution during SMAC mode, see 7.3.7. |

**Table 7-3/G.9954 – MAP information**

| Field name | Field size [bits] | Description |
|---|---|---|
| • • SMAC_EXIT | 1 | Exit from synchronous MAC mode.<br>This flag is used by the master to indicate its intention to end its role as master and to stop sending MAP frames. Endpoint nodes interpret this flag as an indication to exit from SMAC mode and to enter AMAC mode. An endpoint device shall ignore the contents of the TXOP table in a MAP whose SMAC_EXIT flag is set. The endpoint device shall enter into AMAC mode at the end of the current MAC cycle.<br>0 Remain in SMAC mode.<br>1 Exit SMAC mode. |
| • • AMAC_ DETECTED | 1 | Master detected existence of a device operating in AMAC mode. The method used by the master to detect AMAC nodes is implementation dependent.<br>0 Device operating in AMAC mode NOT detected.<br>1 Device operating in AMAC mode detected. |
| • • CP Priority Limit | 3 | Highest priority to be used by G.9954 nodes for transmissions in contention period (CTXOP). May be controlled in order to give priority to CF TXs in an environment (e.g., mixed G.9951/2 and G.9954 network) where CF and CP TXs may collide. Defined values are:<br>0..7 Priority levels |
| • • MAP_IFG | 6 | Size of Inter-Frame Gap (IFG) planned between TXOPs by the master.<br>MAP_IFG silence shall be guaranteed by each endpoint at the end of its TXOP. MAP_IFG is measured in µs and is defined in the range CS_IFG (29) to 63 µs. |
| • • Reserved | 16 | Reserved for future use. Shall be sent as 0 and ignored by the receiver. |
| • Reserved | 32 | Reserved for future use. Shall be sent as 0 and ignored by the received. |
| • SequenceNumber | 16 | MAP sequence number. Modulo counter that is incremented each MAC cycle. |
| • NumTXOPs | 16 | Number of Entries in allocation map.<br>Normally, the minimum number of entries in a MAP is 2 (one entry for the subsequent MAP and the second entry for the UNALLOCATED TXOP). When the SMAC_EXIT flag is set, the number of entries in the MAP may be zero.<br>The maximum number of entries is limited by the maximum size of the MAP control frame as described above. |
| TXOP_TABLE | N × 32 | Variable length table of TXOP descriptors – where N is defined by NumTXOPs. |
| • TXOP[1] | 32 | The encoding of the TXOP descriptor appears immediately below: |
| • • Reserved | 1 | Reserved for future use. This field shall be set to zero by the transmitter and ignored by the receiver. |

**Table 7-3/G.9954 – MAP information**

| Field name | Field size [bits] | Description |
|---|---|---|
| • • TXOP_Length | 15 | The length of the TXOP in units of TIME_SLOTs. TIME_SLOT is defined in 7.3.3.5. |
| • • TXOP_ID | 16 | An identifier used to associate the TXOP with a service flow. The TXOP_ID is a system-wide unique identifier that is composed of the fields described below: |
| • • • SrcDeviceID | 6 | *Device_ID* of device at source of flow. |
| • • • UniqueFlowID | 10 | A unique identifier of the flow within the context of the device at the source of the flow (i.e., *SrcDeviceID).* |
| • … | | |
| • TXOP[N] | 32 | The N'th TXOP described in the TXOP table where N=NumTXOPs. |

The MAP *ControlField* is a collection of flags used to signal control information to endpoint nodes in the network and to control the behaviour of the endpoint nodes. The *ControlField* provides a basic mechanism for the master to distribute policy decisions to endpoint nodes upon detection of certain events. The algorithms used by the master for policy decision making are beyond the scope of this Recommendation.

The *SequenceNumber* may be used to detect lost MAPs and to verify that the "current" media allocation MAP known to a node is "valid" and "up-to-date". The *SequenceNumber* is incremented modulo 16-bits each MAP.

The set of TXOPs planned in the next MAC cycle is described in the TXOP_TABLE. Each table entry contains a TXOP descriptor that specifies the assignment of the TXOP to a device or service (flow) or to a set of devices or services. The assignment of a TXOP is described using the tuple (*SrcDeviceID, UniqueFlowID*). For further information on TXOP addressing, see 7.3.3.4.

The first entry in the TXOP table shall be assigned to the master (if the SMAC_EXIT flag is NOT set) and shall be used for the transmission of the (next) MAP control frame itself.

Since the MAP describes the TXOPs in the next MAC cycle, modifications to the length of the MAP TXOP (i.e., TXOP[1]) actually requires 2 MAC cycles to take effect. This means that a decision to increase the size of the MAP TXOP that occurs in Cycle N shall be described in the MAP of Cycle N + 1 but shall only take effect only in MAC Cycle N + 2.

For further details on the structure of a MAP control frame, see 10.14.1.

### 7.3.3.4 Transmission Opportunities (TXOPs)

The internal structure of a MAC cycle is illustrated in Figure 7-12. It shows an example MAC cycle composed of transmission opportunities (TXOPs) of different types. The following types of TXOPs are defined.

- CONTENTION-FREE TXOP (CFTXOP) – A TXOP allocated to a dedicated (single) network device or service flow.
- CONTENTION TXOP (CTXOP) – A TXOP for which contention-based access is defined amongst a group of network devices or service flows.
- UNALLOCATED TXOP (UTXOP) – An unallocated TXOP is a type of contention-based TXOP where any network device may transmit on a contention basis.

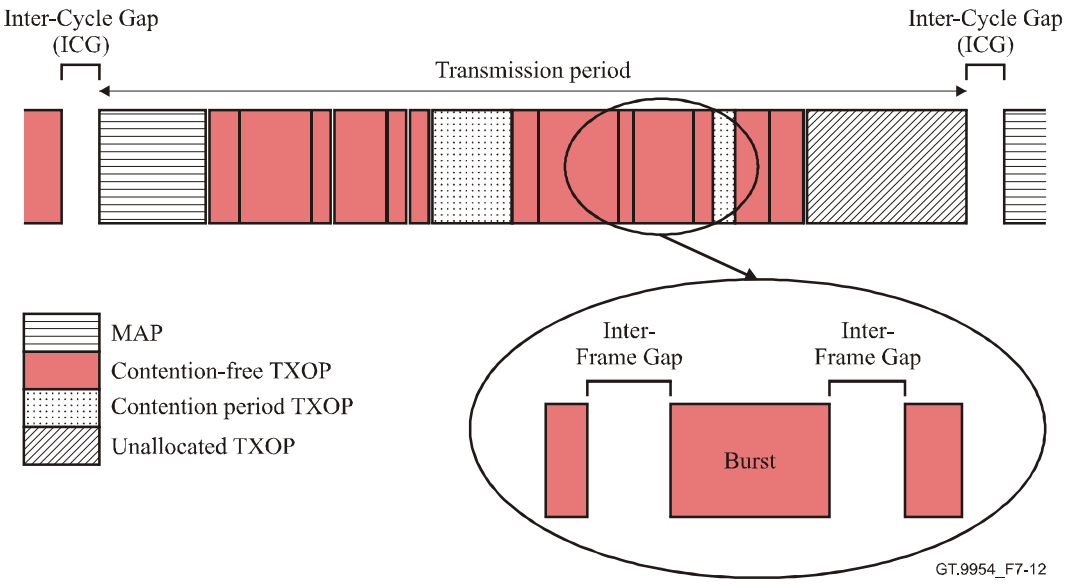


**Figure 7-12/G.9954 – MAC cycle structure**

The *MAP* shall be sent at the beginning of each MAC cycle in the first TXOP of the cycle (as described in the previous MAP). The TXOP used for the transmission of the MAP is, by definition, a Contention-Free TXOP (CFTXOP) and allocated exclusively to the master. It is identified by the MAP TXOP address as defined in 7.3.3.4.2.

The master shall plan media access during a MAC cycle by dividing the available media access time within the MAC cycle time into TXOPs. The master shall allocate Contention-Free TXOPs (CFTXOPs) and Contention TXOPs (CTXOPs) for admitted services requiring QoS contracts. The media access time remaining after all TXOPs have been allocated to specific devices, services or groups shall be assigned by the master as *UTXOPs*. These TXOPs may be used by any device, on a contention basis, for the transmission of non-scheduled traffic, best-effort services, management and network control protocols or traffic for which explicit QoS guarantees are not required.

NOTE 1 – Bandwidth allocated to a network device for transmission may be spread out over a number of TXOPs within the MAC cycle. Although the resource management and scheduling algorithms in the master should attempt to concentrate allocated bandwidth together (in order to reduce the possible number of bursts) it may be necessary to spread the allocation throughout the cycle in order to meet QoS constraints. This may particularly be the case for CBR flows. Similarly, unallocated TXOPs may be scattered throughout the MAC cycle. The placement and length of TXOPs within a MAC cycle are all master scheduler decisions and as such beyond the scope of this Recommendation.

Media access within CTXOPs and UTXOPs shall be performed using contention-based media access methods based on the AMAC mode of operation. Collisions within CTXOPs and UTXOPs, if they occur, are resolved using SMAC collision resolution method (see 7.3.7).

NOTE 2 – Collisions may occur within a CFTXOP in a mixed network of G.9954 and G.9951/2 nodes.

For more information on Collision Resolution schemes, see 7.3.7.

#### 7.3.3.4.1 TXOP Identifiers (TXOP_ID)

All TXOPs defined in the MAP shall be assigned by the master to a single device or flow, on a contention-free basis, or to a group of devices or flows on a contention basis.

The assignment of TXOPs to devices and flows is described by the TXOP identifier or TXOP_ID. The TXOP_ID formed by the tuple (*SrcDeviceID, UniqueFlowID*) where *SrcDeviceID* identifies the device at the source of the flow and *UniqueFlowID* is a unique identifier of the flow within the

context of *SrcDeviceID*. A TXOP shall be assigned to a flow, by the master, during flow setup and reported back to the device at the source of the flow using Flow Signalling Protocol messages.

The TXOP_ID is a 16-bit number that is unique within the network and can have values in the range 0 to 65536.

A device shall only transmit within a TXOP if the device is at the source of a flow to which the TXOP has been assigned. A device may also transmit within certain predefined TXOPs if it contains data that conforms to the semantics of the predefined TXOPs.

#### 7.3.3.4.2 Predefined TXOPs

Predefined TXOPs are special kinds of transmission opportunities that are used for the transmission of messages of a well-defined type or service. Predefined TXOPs are identified by a fixed set of TXOP_IDs. All predefined TXOPs start with a SrcDeviceID component equal to zero. This definition allows for up to 1024 addressable predefined TXOPs.

The values and semantics of predefined TXOPs are implicitly known by all G.9954 nodes. Table 7-4 lists the set of predefined TXOPs:

**Table 7-4/G.9954 – Predefined TXOPs**

| TXOP NAME | TXOP_ID | SEMANTICS |
|---|---|---|
| UNALLOCATED (UTXOP) | 0 | Identifies the unallocated transmission opportunity. This TXOP is available to any device or flow on a contention basis. Any kind of traffic flow can be sent during a UTXOP. |
| Registration (LCP) | 1 | Identifies a contention TXOP reserved for LLC Network Admission Control (Registration) Protocol messages only (see 10.15). |
| Management (LCP) | 2 | Identifies a contention TXOP reserved for LLC Link Control Protocol messages only. |
| G.9951/2 | 3 | Identifies contention TXOPs reserved exclusively for transmission by native G.9951/2 devices. |
| Best-Effort | 4 | A contention TXOP that can be used to transmit data associated with a Best Effort service where a Best Effort service is defined by a traffic flow with a Link Layer Priority 0. For further information on Best-Effort Services see 9.2.5 and 10.6.7.1 on Link Layer Priority Re-mapping. |
| Reserved | 5 ... 1023 | Reserved for future use. |

A G.9954 device shall only transmit frames in a predefined TXOP that conform to the semantics defined for that TXOP and shall NOT use the TXOP for the transmission of any other kind of traffic other than that prescribed for the TXOP.

#### 7.3.3.4.3 Transmission within UTXOPs

Unallocated media access time within a MAC cycle shall be defined in the MAP and assigned, by the master, as an UTXOP using the UTXOP tuple (see 7.3.3.4.2). Any device may transmit within a UTXOP. Devices contend for media access using AMAC media access rules within the time limits (start and end times) of the UTXOP period.

A G.9954 device performing media access within a UTXOP shall cease all transmissions at least MAP_IFG µs before the end of the UTXOP unless it is followed by an adjacent UTXOP. Ceasing all transmissions includes collision resolution signalling if a device is currently involved in a collision resolution cycle. For more information on Collision Resolution during SMAC mode, see 7.3.7.

#### 7.3.3.5 Synchronous MAC protocol timing

Protocol timing parameters under the SMAC mode of operation are the same as for AMAC mode. This includes all the following parameters:

1) InterFrame GAP (CS_IFG);
2) PHY header and trailer length;
3) Minimum PHY frame (minFrameSize);
4) Maximum PHY frame (maxFrameSize);
5) Priority Slot (PRI_SLOT);
6) Signal Slot (SIG_SLOT);
7) Collision Fragment (CD_FRAG);
8) Collision Detection Threshold (CD_THRESHOLD).

For a full description of the AMAC timing parameters, see 7.2.1.1, 7.2.7 and 7.2.7.1. The master shall plan the start-time of TXOP[N] equal to the start-time of TXOP[N – 1] plus the length of TXOP[N – 1].

The length of each TXOP shall include the media time required to transmit actual frame symbols as well as any required *Inter-Frame GAPs* needed to separate consecutive frame bursts. The length of the Inter-Frame GAP used by the master when calculating the length of TXOPs in the MAP shall be signalled to endpoint nodes in the MAP frame (*MAP_IFG).*

NOTE 1 – Normally, a TXOP ends with a MAP_IFG. However, long TXOPs may contain intermediate MAP_IFGs to separate bursts within a TXOP.

A G.9954 device shall not transmit within a TXOP later than $TXOP_{LatesTime} = TXOP_{StartTime} + TXOP_{Length} - MAP\_IFG$ (assuming that the next TXOP is assigned to a different device).

Two consecutive TXOPs with the same TXOP tuple assignment may logically be considered a single TXOP of extended length where the extended length equals the sum of the lengths of the two individual TXOPs. This supports TXOPs with a length greater than the limit imposed by the TXOP Length field in the MAP. In this case, a MAP_IFG is not required between the two consecutive TXOPs and a transmission may extend across the boundary between them.

The transmission time line is divided into time slots of duration TIME_SLOT. TIME_SLOT shall be 500 ns duration. All TXOPs shall start on a TIME_SLOT boundary. The master shall round up the length of TXOPs to integral numbers of TIME_SLOTs when calculating the MAP for a MAC cycle.

All transmissions within a TXOP shall start on a priority slot or signal slot boundary if the MAC is still synchronized with the end of the previous transmission. If the MAC is unsynchronized the transmission may start at any time. In both cases, the transmission must terminate before the end of the TXOP. For a description of synchronized and unsynchronized MAC timing, see 7.2.1.2.

*Example:*

In case a device does not use all of its allocated TXOP time and the start time of the next TXOP is still synchronized with the end of the last transmission, then a G.9954 device must delay its transmission to start on a priority slot boundary.

This can potentially cause a delay (jitter) of up to a maximum of PRI_SLOT – TX_ON μs (i.e., 21 – 4 = 17 μs) as illustrated in Figure 7-13.

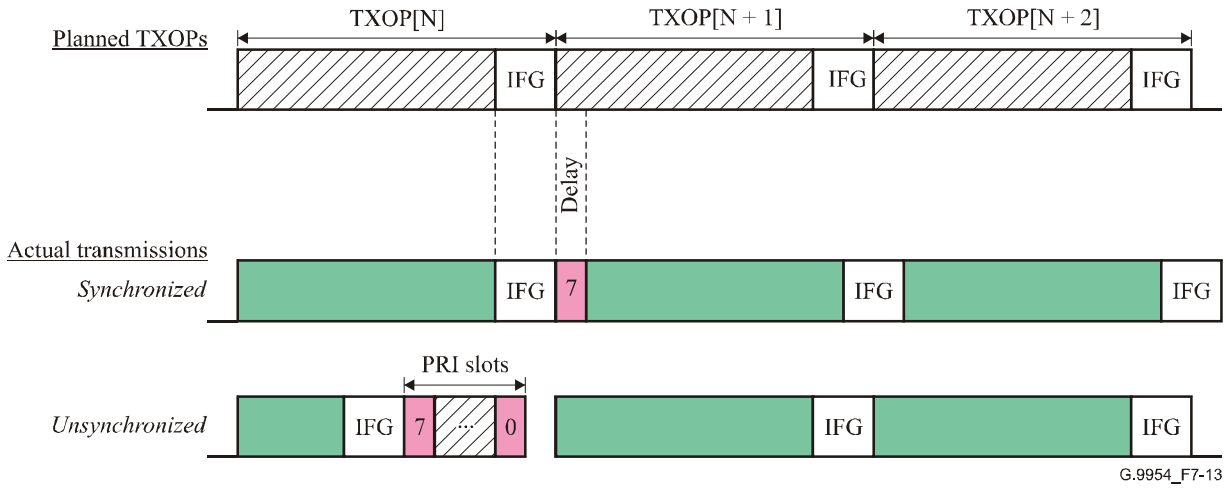


**Figure 7-13/G.9954 – TXOP transmission timing**

The above diagram shows that due to the fact that the device did not use all its TXOP (TXOP[N]), when the time of the next TXOP has arrived, it is no longer positioned at the start of the priority 7 slot, but rather somewhere following the start of priority slot 7. Since transmissions must start on priority slot boundaries (if the MAC is synchronized with the end of the last transmission), the device to which the TXOP is allocated must delay its transmission to the start of the next priority slot boundary. This delay does not accumulate and therefore the maximum potential delay introduced by this event is bounded at 17 µs.

NOTE 2 – It could be argued that the transmission following an unfilled TXOP should be advanced in time to correspond to the priority 7 slot boundary. This, however, would propagate the same condition to all subsequent TXOPs, whereas delaying the transmission to the start of the priority 6 slot has a more localized effect.

### 7.3.3.6 MAC timing synchronization

Network nodes shall synchronize to the master clock reference through the MAP control frame. All timing references specified by the master shall be made relative to the start of the first symbol of the preamble of the MAP frame. This reference point represents offset zero within the MAC cycle.

The current offset within the MAC cycle is reflected in a *synchronous clock counter*. The synchronous clock counter is reset by the arrival of the MAP and counts the progression of TIME_SLOTs relative to the start of the MAC cycle. The synchronization of transmission timing to the start of a TXOP shall be performed using to the synchronous clock counter.

### 7.3.3.7 Propagation delay compensation

Different devices on the network may receive the MAP at different times due to propagation delay. To account for the differences in the propagation delay between stations, the master shall plan for an Inter-Frame Gap (MAP_IFG) in each TXOP that will guarantee, in the worst case, a CS_IFG µs gap between the end of one planned transmission and the start of the next planned transmission accounting for the largest deviation from scheduled TXOP time caused by *propagation delay*.

NOTE 1 – The actual IFG perceived at each station may vary depending on when it received the MAP relative to the master's clock and when the transmission was scheduled by the master. By planning for MAP_IFG and having every station guarantee at least CS_IFG, the effects of propagation delay are bounded and the cycle length will not drift.

The relationship between the minimum guaranteed IFG (CS_IFG), the actual IFG "perceived" by a device and the planned IFG (MAP_IFG) used in the MAP is defined as follows:

$$CS\_IFG \leq IFG \leq MAP\_IFG + 2 \times PD$$

where the parameters appearing in the inequality are described in Table 7-5.

**Table 7-5/G.9954 – IFG parameters**

| Parameter | Description |
|---|---|
| *IFG* | The actual Inter-Frame GAP (IFG) "perceived" by a G.9954 device. |
| *CS_IFG* | The Inter-Frame GAP (IFG) required by the PHY to detect the end of one burst and the beginning of the next burst. |
| *MAP_IFG* | The Inter-Frame Gap used by the master in calculations of TXOP length in the MAP and defined by: $MAP\_IFG = CS\_IFG + 2 \times PD$ |
| *PD* | Maximum propagation delay approximated by transmission at the speed of light (i.e., 300 m equals 1 µs delay). |

The master shall plan for a MAP_IFG gap between bursts when calculating TXOP timing and length and shall advertise the value of MAP_IFG used in its calculations in the MAP. An endpoint device (any endpoint device including the master itself) shall guarantee to terminate its transmission at least MAP_IFG µs before the end of the TXOP.

Figure 7-14 illustrates the variation in perceived IFG from the perspective of different devices in the network in the presence of the effects of propagation delay.

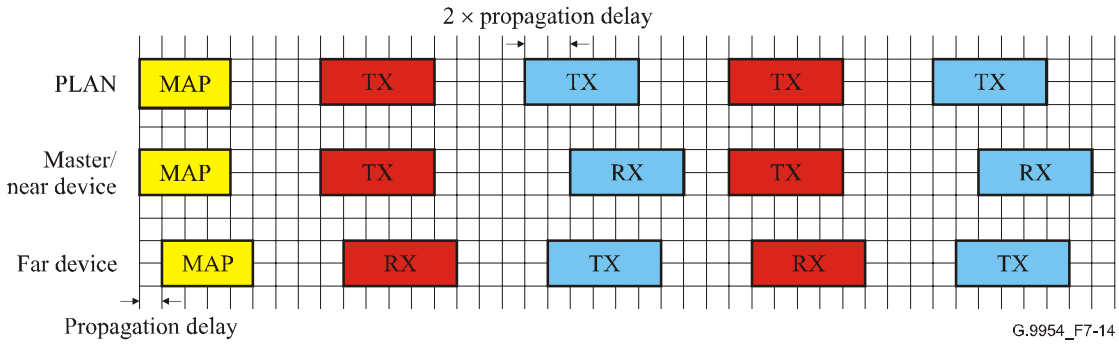


**Figure 7-14/G.9954 – Propagation delay**

NOTE 2 – This mechanism is sufficient to eliminate the need to have non-master devices actually synchronize with the master clock reference. However, for some applications (e.g., voice) it may be important to synchronize with the master clock reference in order to synchronize sampling rates at upper layers. For this purpose, the master distributes its clock using a Link-Layer Timestamp Reporting message, see 10.18.

### 7.3.4 The G.9954 master node functional capabilities

A G.9954 master-capable node is a G.9954 node that, in addition to supporting all of the required capabilities of a G.9954 "endpoint" node, is also able to assume the role of master in the absence of an active master on the network.

In addition to the G.9954 node requirements listed above, a node desiring to be a G.9954 master node shall support all of the following master-related MAC and Link Layer functions:

1) **Network Admission** – Manage the admission of G.9954 nodes requiring QoS guarantees to the network (see 7.3.4.1).

2) **Dynamic Master Selection** – Detect the presence or absence of an operational master on the network and bid for (and assume) the role of network master, if required (see 10.16).

3) **Flow and Bandwidth Management** – Manage the setup, modification and teardown of service flows and the allocation of associated media bandwidth resources in accordance with the services QoS constraints (see 7.3.4.2).

4) **Scheduling** – Plan the media cycle and schedule transmissions such that QoS bandwidth, latency and jitter constraints are met.

5) **MAP Generation and Distribution** – Generate a Media Access Plan (MAP) that represents the output of the bandwidth management and scheduling functions and distribute the MAP each MAC cycle (see 7.3.4.4).

6) **Compatibility Mode Operation** – Detect the presence of G.9951/2 devices operating in the network and adapt network behaviour accordingly (see 8.4).

The G.9954 master shall perform media access using the same media access rules as for Endpoint (non-master) devices and using the same Media Access Plan distributed to the Endpoint devices.

A G.9954 master-capable device shall be able to coexist on the same home network with other (active) master-capable devices. Only a single master device shall exist on the network at any one time. The G.9954 device selected to become master shall be performed automatically using the G.9954 Link Layer master selection protocol (see 10.16).

NOTE 1 – It is NOT a requirement that every G.9954 device be capable of becoming a master.

NOTE 2 – Much of the distinguishing behaviour between different G.9954 master implementations is defined by the Bandwidth Management and Scheduling policies that it implements. Since these aspects of the master are beyond the scope of this Recommendation, related material appearing in this Recommendation should be considered as informative only.

#### 7.3.4.1 Network admission

A G.9954 device that requires the allocation of fixed media bandwidth shall first "register" with the G.9954 master using the Network Admission protocol (see 10.15).

A G.9954 master shall respond to a REGISTRATION request by checking the requesting devices authorization to join the network. The master shall use the MAC address, sent by the requesting device (in the REGISTRATION request) as the device identifier or key for device authentication. If the requesting device is able to be admitted to the network, the master shall assign a DEVICE_ID and return the assigned DEVICE_ID and any network configuration parameters to the requesting device in the REGISTRATION response. If the requesting device is unable to be admitted to the network, the master shall return a status indicating the reason in the REGISTRATION response.

#### 7.3.4.2 Flow and bandwidth management

The G.9954 master shall maintain state information concerning the allocation of media resources in the network and shall control the admission of new services and the allocation of media resources.

Admission control shall be performed in such a way that minimum data-rate as well as maximum latency, jitter and BER characteristics for existing services are not violated.

The master shall service requests to add/delete network service flows and requests to change service flow characteristics using the G.9954 Link Layer Flow Signalling Protocol.

If a request is made to add a new service flow and the requested level of service cannot be met, the master may offer a downgraded level of service. The downgraded service level offered by the master shall not be below the minimum service requirements specified in the service flow specification.

If sufficient resources are not available to accommodate a new service, the master may attempt to reduce the level of service of existing services to their minimum requirement thresholds. If sufficient resources are still unavailable, a denial of service status shall be returned to the requestor.

Denial of service means that no QoS contracts can be given to a particular service. In this case, media access may still be performed on a priority-basis within contention-based TXOPs (CTXOPs).

Similarly, if, for example, changes in line conditions over a logical channel results in a reduction in available network capacity and violation of QoS constraints for admitted services, the master may downgrade service-levels through their allowable limits in order to attempt to accommodate all admitted services. If QoS service constraints, for an admitted service flow, can no longer be met, the master shall notify the source device of the service violation using the Flow Signal Protocol.

Changes in line conditions are actually detected through Rate Negotiation between devices at the endpoints of a channel. If the line conditions change and the transmitter is forced to use a different Payload Encoding (PE), the master will be notified by the transmitting device using the Flow Modification Signalling Protocol. The master should then recalculate media bandwidth reservations to account for the change in PE.

For further information on Quality of Service and details of the protocols used to add new services and modify and remove existing services in the network, see 9.4 and 10.1.2.

#### 7.3.4.3 Scheduling

The G.9954 master shall be capable of allocating media transmission opportunities to services in such a manner that a G.9954 device transmitting within the assigned transmission opportunity will meet QoS bandwidth, latency and jitter constraints for the admitted service flows.

The scheduler shall be responsible for balancing the demands for media bandwidth defined by the traffic specifications of the various admitted services with the total amount of available bandwidth. The output of the scheduling process shall be a Media Access Plan (MAP) that defines the allocated transmission opportunities for the various service flows.

For each admitted service flow, the master scheduler shall calculate the TXOPs required by the service, the start time of the TXOP and the TXOP length. The output of the master scheduling process shall be used to generate the Media Access Plan (MAP).

The G.9954 master shall guarantee the allocation of a minimum amount of unallocated media time (UTXOPs) for the transmission of best-effort traffic and network management and control frames. The minimum amount of reserved transmission time that can be used for these purposes shall be MIN_UTXOP_TIME. This time may be spread over several UTXOPs although no UTXOP shall be less than MIN_UTXOP_LENGTH.

The scheduling algorithm is beyond the scope of this Recommendation since different vendor solutions are possible.

NOTE – The scheduling algorithm should aim to deliver deterministic guarantees for CBR (isochronous) services, statistical guarantees to Variable Bit-Rate (VBR) services and no hard guarantees for Best-Effort services.

Interoperability between master and endpoints from different vendors is guaranteed through the MAP mechanism although QoS results may vary between solutions.

As a benchmark for scheduler performance, a certified G.9954 scheduler should be able to successfully generate a MAP (solution) for a set of scenarios, derived from the QoS requirements as defined in Quality of Service (QoS) parameters for a given maximum PHY rate: MAP Generation and Distribution.

#### 7.3.4.4 MAP generation and distribution

The G.9954 master shall generate and distribute a Media Access Plan (MAP) each MAC cycle.

A new MAP is generated each cycle although the table of TXOPs in the MAP should change only after a change in scheduling decisions as a result of the addition, removal or modification of a service flows or if network conditions change.

The G.9954 master shall distribute the MAP by broadcasting the MAP control frame to all nodes in the network. The MAP control frame shall be broadcast using the most robust Payload Encoding (PE = 33, Spectral Mask 2, 2 Mbaud, 2 bits per symbol).

For further details on the MAP and the structure and timing of the MAC cycle, see 7.3.1 and 7.3.3.2. For further information on the MAP control frame, see 10.14.1.

### 7.3.5 G.9954 endpoint node requirements

A G.9954 endpoint node shall be capable of operating in SMAC mode in the presence of a G.9954 master.

As a minimum requirement, a G.9954 endpoint node shall support the following MAC functions:

1) MAC Cycle Synchronization – A G.9954 endpoint shall synchronize with the master-generated MAC cycle in a master-controlled network.
2) Synchronized Transmissions – A G.9954 endpoint node shall comply with the transmission directives in the current MAP and guarantee that it shall only transmit within a TXOP that is allocated exclusively to it (CFTXOP) or to a group to which it belongs (CTXOP) or within an Unallocated TXOP (UTXOP).
3) Collision Resolution – In the event of media access collisions, a G.9954 node shall be capable of participating in collision resolution according to the rules defined in 7.2.5 and 7.3.7.
4) AMAC Mode Operation – A G.9954 endpoint node shall be capable of operating according to the AMAC protocol, described in 7.2 above, in the absence of a G.9954 master on the network.

Such a device described above is unable to reserve bandwidth for its own transmissions but respects bandwidth allocations to other devices. It is able to synchronize with the MAP and to bound its own transmissions strictly within allocated UTXOPs.

NOTE – The above minimum requirement represents the core functionality upon which the higher-level protocol functions (e.g., registration, flow setup, etc.) can be bootstrapped.

In order to support QoS contracts of bandwidth reservation for flows, a G.9954 endpoint device shall support the following G.9954 MAC and Link Layer functions.

1) Registration – Once an endpoint node has synchronized with the master, the endpoint must perform REGISTRATION. REGISTRATION is the process whereby an endpoint requests entry to the network and, if authorized, is supplied a network address and network configuration data.
2) Flow Signalling – In order to manage QoS flows, an endpoint shall support the Flow Signalling Protocol. The Flow Signalling Protocol is used to set up, modify or tear down flows.

#### 7.3.5.1 Synchronization

A G.9954 endpoint node shall synchronize with the master-generated MAC cycle by detecting the existence of a MAC Media Access Plan (MAP) transmission. Upon detection of a MAP control frame, a G.9954 endpoint node shall reset its synchronous clock counter to zero at the time corresponding to the arrival time of the first symbol of the preamble of the MAP transmission at the wire-interface in the receiver. A G.9954 endpoint device shall schedule its synchronous transmissions within the MAC cycle according to the synchronous clock counter.

If a G.9954 node fails to receive a MAP transmission for SYNC_TIMEOUT ms, or it receives a MAP with the SMAC_EXIT indicator set, the G.9954 endpoint node shall switch to AMAC mode operation.

When operating in AMAC mode, upon detection of a subsequent MAP transmission, a G.9954 endpoint node shall switch to SMAC mode of operation. The mode switch shall occur within MAC_MODE_SWITCH_TIMELIMIT time-units.

#### 7.3.5.2 Synchronized transmissions

When operating in SMAC mode, a G.9954 endpoint device shall perform media access according to the current active Media Access Plan advertised by the master. It shall transmit only within a TXOP assigned exclusively to it or to a group to which it belongs, as defined in 7.3.3.4.1.

A G.9954 endpoint node shall accurately schedule its synchronous transmissions using the synchronous clock counter and comply with synchronous timing constraints specified in 7.3.3.5 and 7.3.3.6.

#### 7.3.5.3 Registration

A G.9954 endpoint node shall register with the master, using the REGISTRATION protocol, if it requires QoS guarantees for services for which it is the source. It shall perform the "registration" sequence once per master session.

A master session commences with the transmission of the first MAP frame by the master and ends after SYNC_TIMEOUT msec without a MAP transmission or upon indication of SMAC_EXIT by the master in the MAP.

A G.9954 endpoint node shall transmit REGISTRATION protocol messages either within a UTXOP or in a REGISTRATION TXOP (see 7.3.3.4.2).

NOTE – Endpoint devices may initially contend for access to the REGISTRATION opportunity. Collisions may be handled by the G.9954 collision resolution methods and/or by retrying after a random number of admission opportunities.

A G.9954 endpoint node shall notify the master of its assigned MAC address in the REGISTRATION protocol message.

Authentication is part of the REGISTRATION process and may be performed by checking that the device, identified by the endpoints MAC address, is authorized to join the network. The authorization procedure is implementation dependent.

The G.9954 endpoint device shall use the DEVICE_ID assigned to it by the G.9954 master in subsequent Flow Signalling Protocol sequences.

For further details on the REGISTRATION protocol, see 10.15.

#### 7.3.5.4 Flow signalling

A G.9954 endpoint node shall support the Flow Signalling Protocol if it supports flows with varying QoS parameters. Support for Flow Signalling is relevant in both SMAC and AMAC modes and should be supported by nodes at both the source and destination of a flow.

In a master-controlled network, the G.9954 endpoint at the source of a QoS Contract flow shall inform the master of Flow Setup, Modification and Teardown requests. It shall notify the master of negotiated rate changes between a source and destination device (i.e., changes in the flows current PE) over a logical channel using the Flow Modification protocol.

For further information on Flow Signalling and Rate Negotiation, see 10.17 and 10.4 respectively.

#### 7.3.5.5 Endpoint processing and scheduling

G.9954 endpoint nodes require little local intelligence in order to schedule transmissions in a master-controlled G.9954 network. Scheduling can be performed solely based on the directions provided in the received MAP. QoS scheduling intelligence is concentrated in the master and expressed in the MAP.

Endpoint nodes may exercise local scheduling intelligence by reassigning the association of services to its allocated transmission opportunities at its own discretion. In other words, services that are allocated specific transmission opportunities by the master may be reassigned, by the endpoint device, to other services, if it so desires.

If local scheduling is performed, the resulting QoS achieved for services originating from the endpoint shall be no worse than that which would be achieved using the master schedule alone.

NOTE – "worse" here is measured in terms of QoS Throughput, Latency, Jitter and BER.

### 7.3.6 Synchronous MAC protocol transmission rule summary

Following is a summary of the G.9954 Synchronous MAC protocol media access and transmission rules:

- MAP Currency – In a master-controlled network, a G.9954 node shall not transmit unless it holds a "current" MAP. A MAP is "current" from the beginning of the MAC cycle to which it refers until the end of the same MAC cycle. See 7.3.3.1.
- Contention-Free TXOP – A G.9954 node shall NOT transmit during a transmission opportunity that is allocated exclusively to another node. See 7.3.3.4.1.
- Predefined Contention Period TXOP – A G.9954 node shall ONLY contend for media access and transmit in a predefined Contention Period TXOP if and only if it intends to transmit a message of the specified service type. Examples of predefined multicast TXOPs include registration slots, bandwidth request slots, etc. See 7.3.3.4.2.
- Contention Period TXOP – A G.9954 node shall NOT contend for media access in a transmission opportunity allocated to a group to which it is NOT a member. See 7.3.3.4.1.
- Transmission Limits – A G.9954 node shall NOT transmit beyond the end of the transmission opportunity assigned to it when operating in a homogeneous G.9954 network. A G.9954 node in a mixed G.9951/2 and G.9954 node network may only exceed the limits of an assigned TXOP if it is involved in a collision resolution process. The start of a TXOP in a G.9951/2/3 mixed node network may shift due to G.9951/2 interference and collision resolution methods. For more information on operation in mixed network, refer to clause 8.
- Collision Resolution – In the event of a collision, G.9954 shall perform collision resolution according to the rules described in 7.3.7.

### 7.3.7 Collision resolution during SMAC mode

Collisions may occur in SMAC mode due to contention with G.9951/2 or G.9954 nodes during a contention period or due to unscheduled transmissions with PNT devices (typically G.9951/2 devices) operating in AMAC mode. G.9954 devices operating in SMAC mode shall detect collisions and participate (if desired) in a collision resolution (CR) process.

The collision resolution method used by G.9954 nodes in SMAC mode shall be determined by the master and signalled to endpoints through the *Collision Resolution Method* control field in the MAP. Two collision resolution methods are defined:

1) DFPQ (G.9951/2-style collision resolution as defined in 7.2.5);
2) Bounded-DFPQ.

DFPQ shall be used in AMAC mode and during operation in a mixed network of SMAC and AMAC nodes (see 8.6.2).

Bounded-DFPQ collision resolution shall be used in SMAC mode in a native G.9954-only network and is based on an adaptation of the DFPQ method used in AMAC mode. The adaptation involves bounding the DFPQ collision resolution cycle within the bounds of the CTXOP in which the collision occurs. To bound the collision resolution cycle within a CTXOP, DFPQ BL/MBL counters are reset to zero at the end of the CTXOP. Collisions should not normally occur during

transmissions within a CFTXOP however, in the event that they do occur collision resolution proceeds as in standard DFPQ and the collision resolution cycle proceeds to conclusion (i.e., BL/MBL counters reach a value of zero).

NOTE 1 – Further consideration should be given to a method based on DFPQ that suspends the value of the BL/MBL counters at the end of a CTXOP and resumes the suspended values at the beginning of the next CTXOP. This method may have the advantage of more tightly bounding collision resolution in the event that the collision resolution cycle does not complete by the end of the CTXOP. Such a method is for further study.

These principles are illustrated in Figure 7-15.

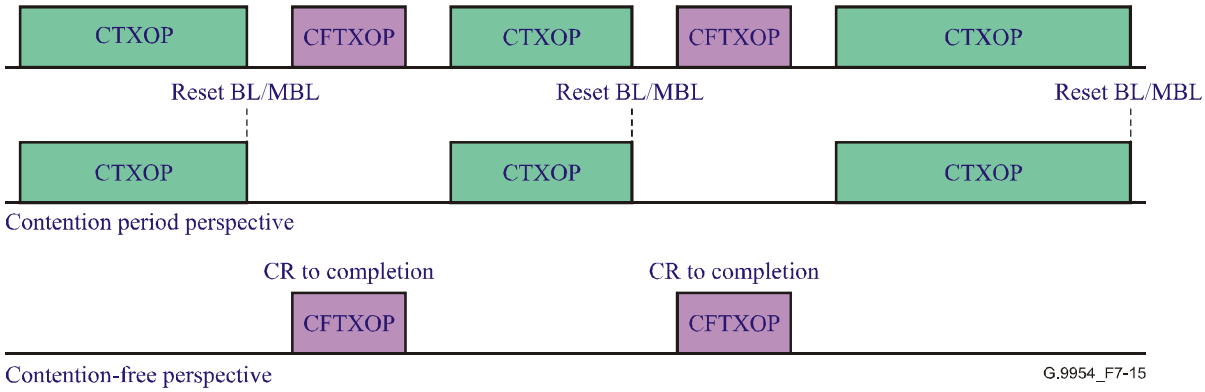


**Figure 7-15/G.9954 – Suspended collision resolution**

Another way of looking at contention periods and collision resolution in SMAC mode is to view a CTXOP as a "bounded" period of G.9951/2-style asynchronous transmissions.

All G.9954 stations may transmit in the contention period (CP) also at priority 7. Each station shall ensure that its frame transmission time plus the MAP_IFG shall not exceed the CP planned time. When a collision occurs, G.9951/2 collision resolution (DFPQ) process takes place. The minimum length of a CTXOP (CP_MIN) shall be $CD\_THRESHOLD + CS\_IFG + 3 \times (SIG\_SLOT) = 217\mu s$, thus ensuring that the required time after collision will always fit in the CTXOP. When a station transmission is shorter than CP_MIN, it shall start its transmission no later than CP_MIN from the CTXOP end. In a native G.9954 network, a station shall not send a signal unless there is sufficient time in the CTXOP after the signal slots for its frame and MAP_IFG.

NOTE 2 – The method of collision resolution described above makes the line activity, as viewed from the perspective of a G.9951/2 node, effectively the same as for regular G.9951/2 transmissions. Although reserved media time for 3 signal slots at the end of a CP are not necessarily required, they guarantee that media timing will look like G.9951/2-compliant transmissions in the event of a collision at the end of the CTXOP.

### 7.3.8 Synchronous MAC parameters

This clause defines SMAC parameters, to supersede any other value of these parameters in other parts of this Recommendation.

**Table 7-6/G.9954 – SMAC parameters**

| Clause | Parameter | Min | Max | Units |
|---|---|---|---|---|
| 7.1 Modes of Operation<br>7.3.5.1 Synchronization | MAC_MODE_SWITCH_ TIMELIMIT | | 50 | milliseconds |
| 7.3.3.1 The MAC cycle | MAP_IFG | CS_IFG | 63 | microseconds |
| 7.3.3.1 The MAC cycle | CS_ICG | CS_IFG | | microseconds |
| | TXOP_LENGTH | 0 | 32767 | TIME_SLOTS |
| 7.3.3.2 The MAC cycle length | CYCLE_MAX | | 50 | milliseconds |
| 7.3.3.2 The MAC cycle length | CYCLE_MIN | 5 | | milliseconds |
| 7.3.3.5 Synchronous MAC protocol timing | TIME_SLOT | 500 | 500 | nanoseconds |
| 7.3.4.3 Scheduling | MIN_UTXOP_TIME | 500 | | microseconds |
| 7.3.4.3 Scheduling | MIN_UTXOP_LENGTH | 217 | | microseconds |
| 7.3.5.1 Synchronization | SYNC_TIMEOUT | | 150 | milliseconds |
| 7.3.7 Collision resolution during SMAC mode | CP_MIN | 217 | | microseconds |
| 8.4 Master requirements in a mixed network | G.9951/2_TXOP_LENGTH | 168 | | microseconds |

## 7.4 Packet aggregation

G.9954 devices shall support the aggregation of Link Layer Frames (packets) into a single physical layer frame (burst). The purpose of packet aggregation is to reduce overheads associated with the physical layer frames by concatenating packets from the same source and to the same destination into a single burst. Packets aggregated into a burst shall either all belong to the same flow or shall all have a priority greater or equal to the priority of the first packet in the aggregated frame.

Aggregation reduces the per-packet overhead by removing the IFG between aggregated packets and allows sharing of common header data (e.g., DA, SA, etc.). In addition, the low baud and low constellation burst header is shared amongst all aggregated packets.

The aggregation frame format shall use the G.9954 Link-Layer Frame Burst Control frame to encapsulate the aggregated packet data. This Link Layer Control Frame format is described in full detail in 10.13.

Aggregation may be performed under both SMAC and AMAC modes of operation. In either case, aggregation shall conform to the following basic rules:

- The Maximum length of the aggregated frame shall not exceed the maximum allowed time on the wire.
- The maximum number of aggregated frames in a burst shall be negotiated between source and destination, either using the CSA protocol or Flow Signalling Protocol.
- All aggregated frames in a burst shall have the same Source and Destination address. The destination address may be a BROADCAST or MULTICAST address.
- The priorities of all aggregated frames in a burst shall be all greater than or equal to the priority of the first sub-frame of the burst.
- A burst termination header shall be used to indicate the end of a burst.

When operating in SMAC mode, aggregation may be performed up to the limits of the size of the TXOP in which the frame will be transmitted or up to the maximum link-level frame size, whichever is smaller.

Since TXOP length for flows are calculated as a function of flow latency requirements, it could be said that aggregation is performed up to the limits of the allowed latency for a flow.

When operating in AMAC mode, the MAC protocol may use the latency specification and nominal packet size specification for a flow to determine the amount of aggregation to perform.

Figure 7-16 shows a breakdown of the frame aggregation format.

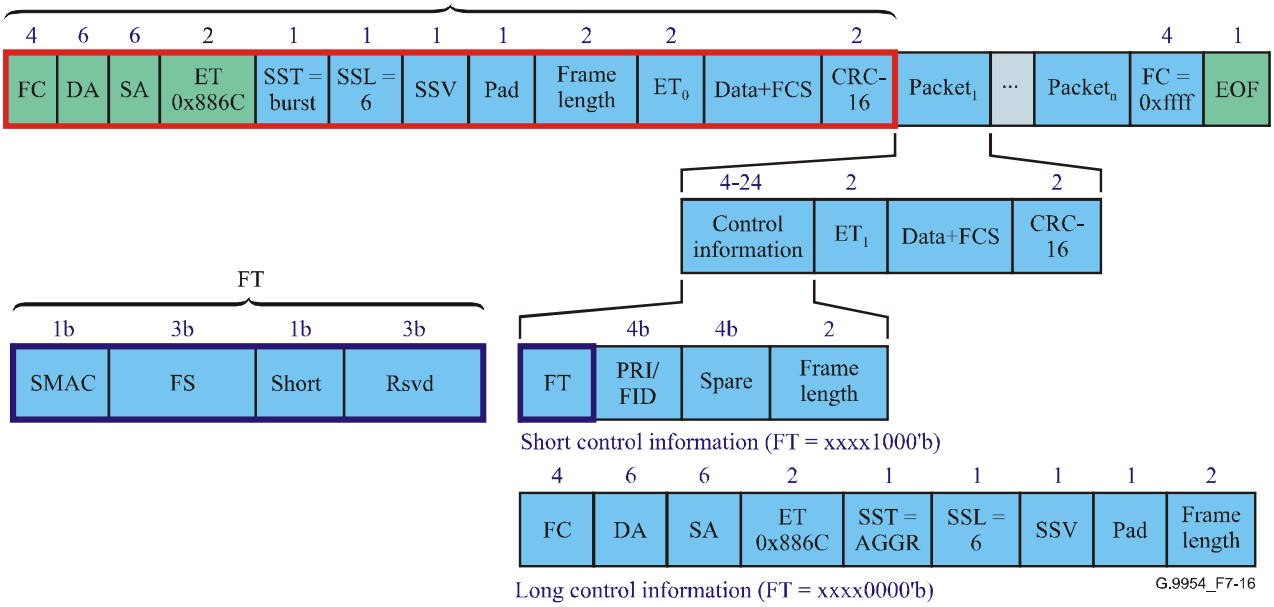


**Figure 7-16/G.9954 – Aggregation frame format**

The aggregation frame format supports a "short" and "long" aggregation form. In the "long" form, each aggregated frame shall contain a full packet header. This form contains redundant header information but allows simple processing, both on transmit and receive. In the "short" form, redundant header data appears only once, in the first packet (packet0) and is subsequently shared by all other aggregated frames. This form is more efficient in its media usage but may involve extra processing capabilities.

The aggregation form supported by a device shall be advertised using the CSA protocol. The "long" aggregation format shall be supported by all implementations. Support for the "short" aggregation format is optional.

For frames using the "long" control information header, the FCS field shall have the same meaning as described in IEEE Std 802.3 frame and is calculated over the DA, SA, EtherType and Data fields of the frame. For frames using the "short" control information header aggregation form is used, the FCS shall be calculated from the first bit of the FT field through the last bit of the Data fields. The CRC-16 shall be calculated over the same respective fields.

# 8 Compatibility specification

## 8.1 Spectral compatibility with other services on the same wire

The PSD mask specified is such that compliant transmitters should be able to meet FCC Part 68 Section 308-e-1-ii.

The mask also specifies a limit of –140 dBm/Hz below 2.0 MHz, which ensures compatibility with ITU-T Recs G.992.1 and G.992.2 and ISDN.

The mask includes notches covering the Radio Amateur bands (e.g., between 7.0 and 7.3 MHz), which reduces the maximum PSD to –81.5 dBm/Hz. This is lower than the VDSL recommendations for PSD in the amateur bands. Since the VDSL spectral compatibility has been developed over the last several years in several standards bodies, including the ITU-T, this spectral mask should be compatible with RFI emission requirements in countries outside North America, such as UK, Japan, Germany and France.

## 8.2 Coexistence and interoperability with G.9951/2 and AMAC nodes

G.9954 is inherently backward compatible with G.9951/2 as it uses the same PHY header, frame format and protocol timing parameters as G.9951/2. Although higher baud payloads are supported in G.9954, the baud parameter is negotiated between transmitter and receiver. On the wire, transmissions from G.9954 nodes operating in SMAC mode appear like standard G.9951/2 transmissions, albeit at higher baud and using Frames that are (forward compatible) but possibly unrecognized by G.9951/2 nodes.

In a mixed network of G.9951/2 and G.9954 nodes, the problem of compatibility is reduced to one of coexistence of the SMAC and AMAC protocol modes operating simultaneously on the network. In a homogeneous network of G.9954 nodes in SMAC mode, all media access timing is planned and collisions may only occur during controlled contention periods. However, in a mixed network of G.9951/2 and G.9954 nodes, no such guarantees exist. G.9951/2 node transmissions may collide with planned transmissions, they may burst into silence within a transmission opportunity and they may extend beyond the extent of a transmission opportunity.

The rest of clause 8 describes a method of coexistence and interoperability with G.9951/2 devices that preserve the synchronous nature of G.9954 node SMAC mode transmissions while accommodating for unsynchronized G.9951/2 transmissions.

G.9954 nodes shall coexist and interoperate with G.9951/2 nodes in a mixed network of G.9951/2 and G.9954 nodes. Furthermore, a network of G.9954 nodes operating in SMAC shall be able to coexist and interoperate with other PNT devices operating in AMAC mode.

Normally, only G.9951/2 devices will operate in AMAC mode in the presence of a G.9954 master. However, a "rogue" G.9954 node may appear in an environment where the G.9954 node cannot "hear" MAP transmissions.

In the following clauses, the terms "G.9951/2 node" and "AMAC node" may sometimes be used interchangeably.

## 8.3 Detection of G.9951/2 nodes

A G.9954 master node shall be capable of detecting the presence of G.9951/2 nodes on the network.

Upon detection of a G.9951/2 node, the master shall notify G.9954 endpoint nodes by signalling the event using the AMAC_DETECTED flag in the MAP frame.

A G.9954 endpoint device should not transmit to a device using a payload encoding (PE) that is not supported by the device.

The Payload Encoding used to communicate with a device is negotiated through Rate Negotiation and no specific knowledge is necessary as to the PNT version of the receiver. However, the version number of the PNT receiver may be used to select an appropriate initial PE to use before Rate Negotiation completes. This implies that the initial PE for communication with a G.9951/2 node should be Spectral Mask#1, 2 Mbaud, 2 bits per symbol while the initial PE for communication with a G.9954 node may start at a higher rate (e.g., Spectral Mask #2, 8 Mbaud, 2 bits per symbol).

The mechanism used to detect the presence of G.9951/2 nodes on the network is implementation dependent. The master may detect G.9951/2 nodes in the network using version number information in the CSA protocol message or by detecting collisions during CFTXOPs. Collisions during a CFTXOP may be caused by any node operating in AMAC mode and not necessarily an G.9951/2 node.

### 8.4 Master requirements in a mixed network

G.9951/2 device presence shall be signalled if one or more G.9951/2 devices are detected on the network. The absence of a G.9951/2 device shall be signalled if no G.9951/2 devices were "heard" on the network in the last two minutes.

Upon detection of G.9951/2 nodes on the network, the master shall reserve at least one TXOP of length G.9951/2_TXOP_LENGTH and allocate it exclusively for the use of G.9951/2 nodes. The TXOP shall be allocated from otherwise unallocated media resources within the cycle. The master shall reserve one G.9951/2 TXOP for each link-layer priority in use.

NOTE 1 – By allocating a TXOP exclusively to G.9951/2 nodes, this guarantees that:

a) G.9954 nodes will not contend for media access with G.9951/2 nodes during this period; and

b) G.9954 nodes will not attempt to enter a collision resolution cycle started during the G.9951/2 TXOP.

This means that if several G.9951/2 nodes (all of the same priority) are contending for media access at the start of a G.9951/2 TXOP, all the contending G.9951/2 nodes will succeed in accessing the media before a G.9954 node manages to access the media, even if the length of the G.9951/2 TXOP is (relatively) small. This is guaranteed by DFPQ, which ensures that new nodes cannot join an ongoing collision resolution cycle. G.9951/2_TXOP_LENGTH is defined to be large enough to include media time for all priority slots.

The reserved media resources shall appear in the advertised MAP as TXOPs allocated exclusively to G.9951/2 nodes. A TXOP allocated to a G.9951/2 node shall be identified by the predefined G.9951/2 TXOP address identifier (see 7.3.3.4.1).

Upon discovery of G.9951/2 or AMAC nodes on the network, the master may modify its own behaviour as well as that of the endpoints in order to better adapt with the mixed environment. Modifications to the behaviour of endpoint nodes, if required, are signalled by the master to the endpoints through the MAP frame control fields. The decision making process taken by the master is beyond the scope of this Recommendation.

NOTE 2 – A G.9954 device should attempt to compensate for possible interference caused by unsynchronized G.9951/2 and AMAC devices. The measures employed may include allocation of extra bandwidth to TXOPs in order to compensate for latency introduced into the cycle. The amount of spare time added to a TXOP is service dependent but should be sufficient to at least accommodate a whole packet for the particular service.

In addition, upon discovery of G.9951/2 (or AMAC) nodes on the network, the master shall signal, through the MAP, the following changes in endpoint behaviour:

1) Collision Resolution Method –The master shall set the collision resolution method to AMAC mode in a mixed network. For further information on Collision Resolution in a mixed network see 7.2.5 and 8.6.2.

2) Cycle Latency Repair – The master shall signal the policy to be used to control latency at the start of the MAC cycle if the cycle start is delayed. For further information, see 8.6.4.

3) TX Priority Limit in Contention Period – The master shall signal the Link-Layer Priority to be used for all transmissions within a contention-based TXOP (CTXOP or UTXOP).

NOTE 3 – Since transmissions within CFTXOPs are equivalent to transmissions at priority = 7, by setting the TX priority limit in CTXOPs to a value that is less than 7 (e.g., priority = 6), priority preference can be given to contention-free transmissions over contention-based transmissions. This approach should be adopted.

## 8.5 Transmissions to G.9951/2 nodes

Transmissions to G.9951/2 nodes from G.9954 nodes shall conform to the G.9951/2 frame format. Transmissions shall be performed using a Spectral Mask #1 Payload Encoding. The PE baud and bits-per-symbol shall be negotiated between the G.9951/2 and G.9954 node using the standard PNT Link Layer Rate Negotiation Protocol.

The Frame Control Frame-Type (FT) of frames sent to a G.9951/2 node shall be zero.

Frames sent by a G.9954 node to the broadcast address shall be sent using a Spectral Mask #1 Payload Encoding in the presence of G.9951/2 nodes on the network. Similarly, frames sent to the multicast address are sent using a Spectral Mask #1 Payload Encoding if a G.9951/2 node is in the set of active multicast listeners. For a definition of "active multicast listeners", see the description of the Rate Negotiation Protocol in 10.4.

A G.9954 node shall NOT send aggregated frames to G.9951/2 node and shall assume that "Frame Bursting" is NOT supported. For further information on Frame Bursting, see the CSA and Frame Bursting Protocols in 10.13 and 10.6 respectively.

## 8.6 Coexistence of Synchronous and Asynchronous MAC modes

In a mixed network, G.9954 nodes operating in SMAC mode shall continue to operate under master control while accommodating for interference that may be introduced by unscheduled AMAC mode transmissions.

Accommodation for unscheduled (asynchronous) transmissions in SMAC mode shall be performed using a combination of carrier sensing and collision detection techniques by all nodes. G.9954 synchronous mode transmissions shall compete for media access with asynchronous mode transmissions, as priority 7 frames. Collisions shall be resolved using the collision resolution method defined for the AMAC mode.

### 8.6.1 Planned transmissions and carrier sensing

A G.9954 node operating in SMAC mode shall not transmit within a scheduled TXOP unless all of the following conditions hold:

1) TXOP is allocated to the G.9954 node (or to a group to which the G.9954 node belongs);
2) TXOP start time has arrived (as measured from the start of the MAP);
3) Media is IDLE;
4) CS_IFG gap has completely elapsed;
5) BL/MBL counters are zero.

If the first two conditions hold but the media is sensed to be BUSY or a CS_IFG gap has not been sensed, the transmission shall be contained (withheld) until all conditions are met.

NOTE – The above requirement implies that carrier sensing is required at all times, even under SMAC mode of operation.

### 8.6.2 Collision detection and resolution

In a mixed network of G.9951/2 and G.9954 nodes, collisions may occur in the same manner as in a network operating in AMAC mode. G.9954 nodes operating in SMAC mode shall contend for media access with G.9951/2 nodes at priority 7.

Collisions, if they occur, shall be resolved using AMAC collision resolution method as follows:

If a collision occurs during a transmission in a Contention-Free Period (CFTXOP), the transmitting node shall enter into a Collision Resolution Cycle and contend for media access until transmission succeeds. The transmission may occupy up to the full length of time allocated to the CFTXOP

irrespective of the actual transmission start time. This may cause the CFTXOP to extend beyond its expected end-time.

If a collision occurs within a Contention-Period TXOP (CTXOPs), transmitting nodes may contend for media access as in the case of a native network of G.9954 nodes. The extent of the CTXOP shall not be extended in this case irrespective of the actual transmission start time. This implies that in compatibility mode, the length of a CTXOP may shrink given a start-time that may shift in time (due to previous unplanned transmissions) while the end-time of the TXOP is fixed. This is intentional and is used to repair latency introduced by unscheduled transmissions (see 8.6.3).

At the end of a CTXOP, if collision resolution is in progress (i.e., BL/MBL collision resolution state counters are non-zero), the counters shall reset to zero with the passage of an empty priority slot. This behaviour is the same as in AMAC mode. This may introduce PRI_SLOT latency in the start of the following TXOP.

Examples:

Figure 8-1 bellow illustrates the effect of a collision that occurs at $T_1$ on the transmission scheduled in TXOP 1.

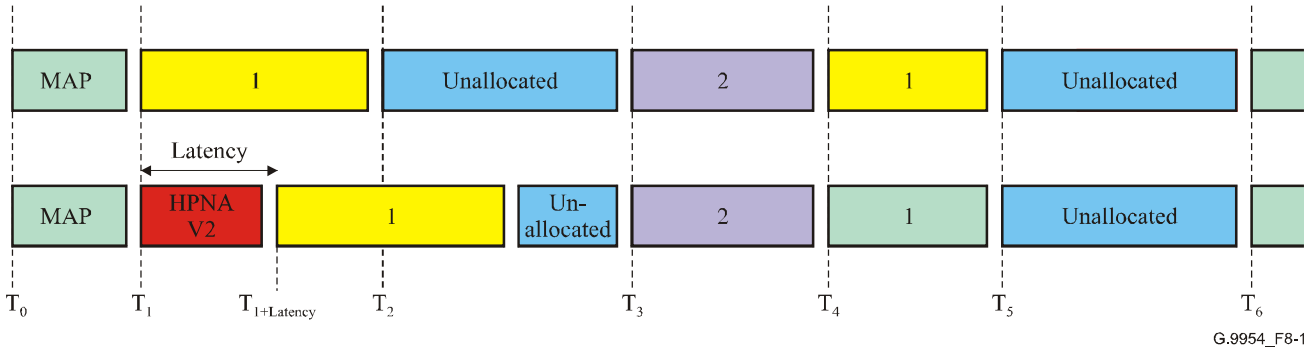


**Figure 8-1/G.9954 – Collision and timing jitter**

Collisions with G.9951/2 nodes may introduce delays in the start of transmission in subsequent scheduled transmissions. Depending upon the number and length of unscheduled transmissions, it is possible that two or more consecutive scheduled transmissions will be contained by carrier sensing until the media becomes IDLE. At this point, all the contained nodes will attempt to access the media together resulting in collisions between the scheduled transmissions. In case of collisions between these scheduled transmissions, regular collision resolution methods shall be applied. There is NO guarantee of the ordering of transmissions after collision resolution and this order may be different to that scheduled in the original MAP. This is illustrated in Figure 8-2.

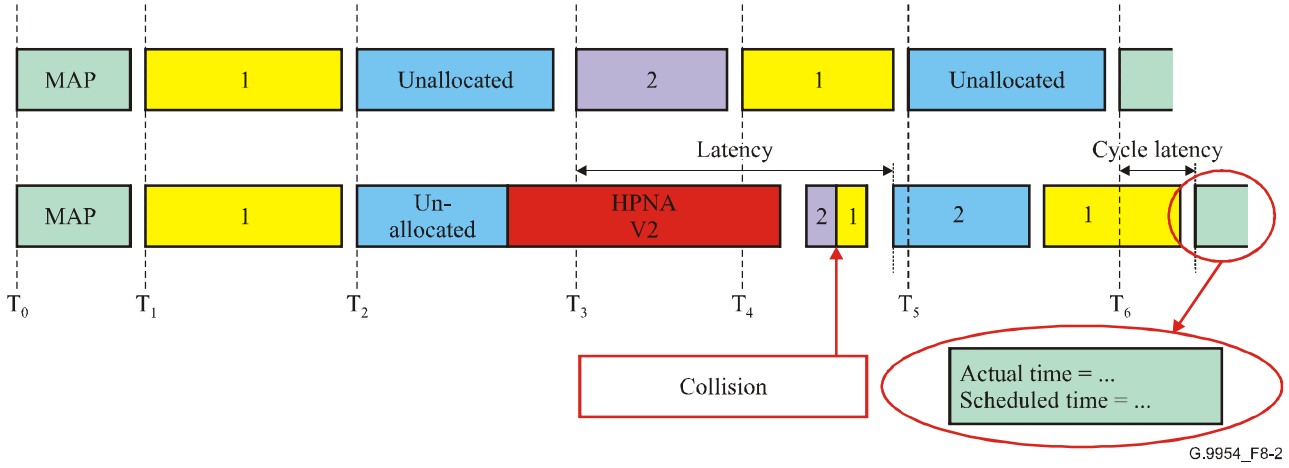


**Figure 8-2/G.9954 – Collisions between scheduled transmissions**

### 8.6.3 Timing compensation (Latency Repair) within a MAC Cycle (Informative)

Timing delays in scheduled (synchronous mode) transmissions, caused by unscheduled G.9951/2 transmissions, are naturally compensated for, if sufficient spare bandwidth (i.e., unallocated TXOPs) exist in the MAC cycle.

Unallocated bandwidth (UTXOPs) has the effect of absorbing delays in the start of scheduled transmission. This occurs because unallocated TXOPs, by definition, do not need to guarantee a fixed amount of bandwidth; rather, they represent the remaining media time. If some of the unallocated media time is consumed by latency in a scheduled transmission, the length of the UTXOP is simply shortened by the amount of time consumed. This means that the end time of a UTXOP is fixed while its start time may shift by the amount of delay introduced in the MAC cycle.

This is illustrated in Figure 8-1 where a delay in the start of the transmission in TXOP 1 is compensated for by allowing the transmission in TXOP 1 to extend beyond its scheduled end time and effectively consume the spare bandwidth following it. Whereas a scheduled TXOP is always fixed in length with varying start and end times, an UTXOP is variable in length and has a fixed end time but a floating start time. This effectively adjusts the size of the unallocated TXOP by the amount of delay (jitter) introduced by previous transmissions.

If the start time of an unallocated TXOP is later than the scheduled end time, the unallocated TXOP is considered to be NULL or non-existent.

### 8.6.4 Timing compensation between MAC cycles

Timing delays that occur within a MAC cycle may propagate throughout the cycle and introduce delays in the start of the next MAC cycle (i.e., delay the next MAP transmission).

Delay (latency) introduced in the start of a MAC cycle can be calculated by each endpoint device by subtracting the Actual Arrival Time of the MAP from the Scheduled Arrival Time. The Actual Arrival Time shall be captured by the receiver. The Scheduled Arrival Time can be calculated from information in the previous MAP by summing the lengths of all TXOPs and adding this to the Actual Arrival Time of the corresponding MAP.

If the Latency Repair method specified in the MAP indicates "adjust clock" a G.9954 endpoint node shall compensate for the MAC cycle jitter by setting their synchronous clock counter, on the receipt of the MAP frame, to the calculated latency in the start of the MAC cycle rather than resetting the synchronous clock counter to 0.

NOTE – Since all transmissions are scheduled relative to the base time (zero), this mechanism can be used to carry over latency compensation from one cycle to the next. Alternatively, no particular corrective action may be taken if sufficient unallocated (spare) bandwidth exists that can be used to empty buffers that have

accumulated samples as a consequence of the latency. It is the responsibility of the master to decide on the appropriate latency repair strategy and signal this information to the network nodes through the MAP.

### 8.6.5 Examples of operation (Informative)

This clause shows, by example, a set of possible G.9951/2 interference scenarios. All the scenarios illustrated assume the following transmission MAP in Table 8-1.

**Table 8-1/G.9954 – Active media access plan**

| TXOP index | Device | TXOP type | Start time | Length |
|---|---|---|---|---|
| 0 | master | Contention-free | $T_0 = 0$ | $L_0$ |
| 1 | "A" | Contention-free | $T_1 = L_0$ | $L_1$ |
| 2 | BROADCAST | Unallocated | $T_2 = T_1 + L_1$ | $L_2$ |
| 3 | "B" | Contention-free | $T_3 = T_2 + L_2$ | $L_3$ |
| 4 | "C" | Contention-free | $T_4 = T_3 + L_3$ | $L_4$ |
| 5 | BROADCAST | Unallocated | $T_5 = T_4 + L_4$ | $L_5$ |

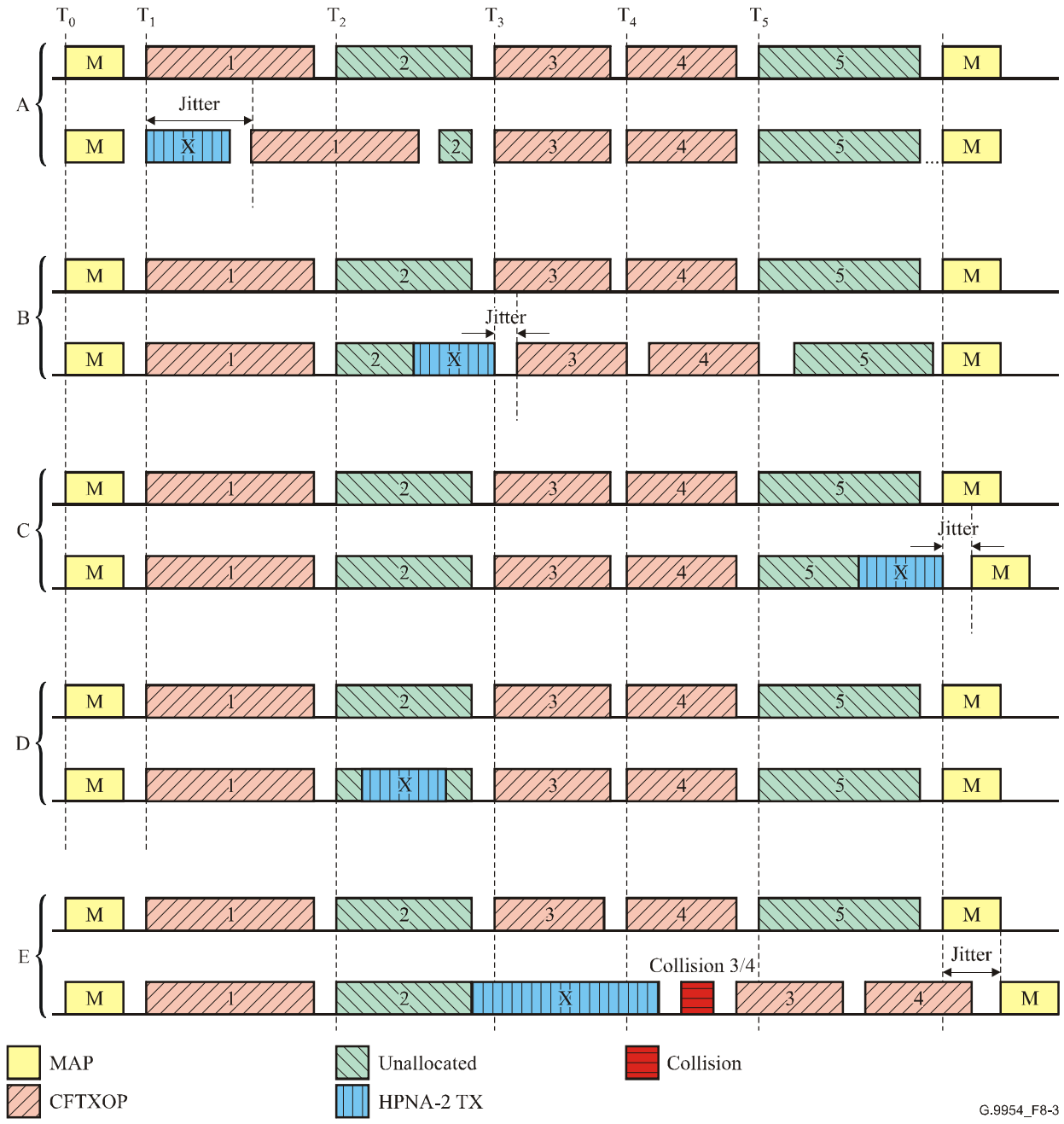


**Figure 8-3/G.9954 – G.9951/2 and G.9954 coexistence scenarios**

In Figure 8-3, example A, the master transmits the MAP at $T_0$, and then an asynchronous node takes control of the medium at $T_1$ before synchronous node "A" begins transmitting. The asynchronous node transmits an unscheduled asynchronous transmission X. Since the asynchronous node is unaware of the synchronous transmission rules, it will transmit as long as it requires. Since all the nodes on the medium support CSMA/CD techniques, the asynchronous transmission is uninterrupted. Using carrier sense techniques, synchronous node "A" waits until transmission X is finished, waits an additional inter-burst gap, and then begins transmission. This leads to jitter as indicated by an arrow. Synchronous node "A" will transmit for the entire allocated time even though TXOP 1 starts late. Since, in this example, this transmission does not extend beyond the end time of unallocated TXOP 2, TXOP 3 begins on time at $T_3$ and the jitter is not propagated.

In Figure 8-3, example B, the master transmits the MAP as scheduled at $T_0$, and then synchronous node "A" transmits during TXOP 1 as scheduled at $T_1$. During unallocated TXOP 2, the medium is

idle and, using carrier sense techniques, an asynchronous node takes control of the medium. The asynchronous node transmits an unscheduled asynchronous transmission X that extends beyond the scheduled time of TXOP 2. Synchronous node "B" uses carrier sense techniques and delays its scheduled transmission (TXOP 3) until transmission X is finished and an inter-burst gap has passed. Then synchronous node "B" begins its transmission, with a delay that leads to jitter as indicated by an arrow. The transmission for TXOP 4 is also delayed by the same amount, because synchronous node "B" will transmit for the entire allocated time of TXOP 3. Since, in this example, the transmission for TXOP 4 does not extend beyond the end time of unallocated TXOP 5, transmission of the MAP of the next cycle begins on time and the jitter is not propagated.

In Figure 8-3, example C, the master transmits the MAP as scheduled at $T_0$, and then synchronous node "A" transmits during TXOP 1 as scheduled at $T_1$. Similarly, synchronous node "B" transmits during TXOP 3 as scheduled at T3, and synchronous node "C" transmits during TXOP 4 as scheduled at $T_4$. During unallocated TXOP 5, the medium is idle and an asynchronous node takes control of the medium. The asynchronous node transmits an unscheduled asynchronous transmission X that extends beyond the scheduled time of TXOP 5 ($T_5 + L_5$). The master uses carrier sense techniques and delays its scheduled MAP transmission until transmission X is finished and an inter-cycle gap has passed. Then it begins transmitting the next MAP, with a delay that leads to jitter as indicated by an arrow. The next MAP will be used to determine the amount of jitter introduced. The synchronous nodes may set their own clocks to $T_0$ + jitter when receiving the MAP delayed by the jitter. This will enable the synchronous nodes to attempt to transmit on time, and not delayed by the jitter. For example, if the MAP is followed by an unallocated TXOP 1', and then a unicast TXOP 2', then if the duration of unallocated TXOP 1' is more than the jitter, the subsequent TXOPs of the cycle will start on time. This mechanism enables compensation for jitter in the transmission of a MAP.

In Figure 8-3, example D, the master transmits the MAP as scheduled at $T_0$, and then synchronous node "A" transmits during TXOP 1 as scheduled at $T_1$. During unallocated TXOP 2, the medium is idle and an asynchronous node takes control of the medium. The asynchronous node transmits an unscheduled asynchronous transmission X that, in this example, does not extend beyond the scheduled time for TXOP 2 ($T_2 + L_2$). No jitter is introduced into subsequent transmissions, and therefore no accommodation needs to be made.

In Figure 8-3, example E, the master transmits the MAP as scheduled at $T_0$, and then synchronous node "A" transmits during TXOP 1 as scheduled at $T_1$. During unallocated TXOP 2, the medium is idle and, using carrier sense techniques, an asynchronous node takes control of the medium. The asynchronous node transmits an unscheduled asynchronous transmission X that extends beyond the scheduled time of TXOP 2 and into the scheduled time of TXOP 3. Synchronous node "B" uses carrier sense techniques and delays its scheduled transmission until transmission X is finished and an inter-burst gap has passed. However, since the transmission time for TXOP 4 ($T_4$) is also past, synchronous node "C" uses carrier sense techniques and delays its scheduled transmission until transmission X is finished and an inter-burst gap has passed. What results is a collision between synchronous node "B" and synchronous node "C" which is subsequently resolved using collision resolution method at the expense of further jitter in the MAC cycle.

## 9 G.9954 Quality of Service

G.9951/2's support for 8 priority levels provides a basic Quality of Service (QoS) mechanism for differentiating between different kinds of services. This mechanism is compatible with several existing mechanisms for differentiating between classes of service such as the IEEE 802.1D recommendations for the VLAN Priority Tag (IEEE 802.1P) and the PRECEDENCE bits defined in the original interpretation of the Type Of Service (TOS) field found in an IP packet header using the Differentiated Services (Diffserv) protocol.

Although priority classification of services provides some level of QoS support, it cannot provide QoS guarantees with strict latency and jitter budgets. In order to provide strict QoS Contracts, the G.9954 MAC provides a mechanism based on the concept of *flows* that is compatible with an RSVP-like protocol and supports the specification of QoS in terms of explicit traffic and rate parameters and not just a relative ordering of packets. Traffic shaping, scheduling and policing mechanisms, based on these well-defined QoS parameters, are subsequently used to provide strict control over network throughput, latency and jitter performance.

This clause specifies the G.9954 QoS solution.

## 9.1 General description

The G.9954 QoS mechanism is based on the concept of a *data flow* (or *flow* for short). A flow represents a unidirectional flow of data between network nodes based on well-defined QoS traffic and rate parameters that allow strict control over network throughput, latency, jitter and BER parameters.

Flows are set up and torn down on a service-by-service basis. The G.9954 master is responsible for allocating bandwidth for flows, upon request, and for advertising the bandwidth allocation decision in the Media Access Plan (MAP). Network nodes are responsible for scheduling their transmissions according to the constraints of the advertised MAP.

The bandwidth allocation algorithm shall enforce and guarantee QoS parameters. Consequently, bandwidth reservation requests associated with a flow setup are subject to admission control policing and shaping by the master. Flow setup requests that cannot be met according to the requested parameters are either rejected or QoS parameters are re-negotiated by the master.

Bandwidth requirements for a flow may be modified throughout its life time in order to more effectively support changing bandwidth requirements that are characteristic of "bursty" and variable bit-rate (VBR) data streams and changing line conditions.

Flows are set up by convergence layers, either implicitly – upon identification of a new service, explicitly – in response to higher-level protocol messages (e.g., RSVP reservation requests) or upon network admission according to a predefined specification/configuration. Flows may similarly be torn down implicitly, upon detection of inactivity or explicitly upon termination of a service, in order to free network resources associated with the flow.

It is the responsibility of the convergence sub-layer to map incoming data streams onto the appropriate flow that meet their individual QoS requirements.

In summary, the main QoS features supported by G.9954 MAC protocol are as follows:

- Statistical and deterministic QoS guarantees for bandwidth, jitter, latency and BER;
- *Traffic classes* and *service flows* described by well-defined traffic and rate parameters;
- Constant and variable bit-rate flows;
- Flow management including flow admission control, resource reservation, QoS negotiation/re-negotiation, flow setup and tear-down;
- Frame classification based on traffic filter specification, e.g., IP TOS, VLAN priority tag, protocol type, source/destination address, etc.;
- QoS flow policing, shaping and scheduling.

## 9.2 Service flows and QoS parameters

A flow describes a simplex communication channel, with well-defined QoS characteristics, between source and destination device. The QoS characteristics of a flow are described by a set of traffic and rate parameters which are communicated between G.9954 devices using the Flow Signalling protocol (see 9.4 for more details).

The QoS characteristics of a flow are defined by the parameters summarized in Table 9-1 and defined in subclauses below.

**Table 9-1/G.9954 – Flow properties**

| Field name | Description |
|---|---|
| Source Address | The MAC address of device at source of the flow |
| Destination Address | The MAC address device that is the destination of the flow (may be the broadcast address) |
| Flow ID | Unique identifier of the flow between the Source and Destination Addresses. The flow ID is assigned by the G.9954 device at the source of the flow |
| Service Class | Identifies a Class of Service (CoS). Used as a short-hand form for specifying a "well-defined" set of QoS parameters without requiring the specification of individual QoS parameters |
| Priority | Link Layer priority assigned to the flow |
| Service Type | Defines the type of service that the flow supports:<br>0 CBR<br>1 rt-VBR<br>2 nrt-VBR<br>3 BE |
| Maximum Latency | Maximum tolerable transmission and queuing delay according to Table 10-69 |
| Maximum Jitter | Maximum delay variation according to Table 10-69 |
| ACK Policy | 0 None<br>1 LARQ |
| FEC Policy | 0 None<br>1 RS<br>2-3 Reserved |
| Aggregation Policy | 0 No Aggregation<br>1 MAC-Level Aggregation |
| CRC Error Handling Policy | 0 Do not discard packets with CRC errors.<br>1 Discard packets with CRC error. |
| Nominal Packet Size | The nominal packet size in octets for packets associated with the service. A value of 0 indicates an unspecified or unknown value. |
| Maximum data rate | Peak burst rate in 4 kbit/s units. Takes into account the net (payload) data rate |
| Average data rate | Average bit rate required by the service in units of 4 kbit/s |
| Minimum data rate | Minimum required bit rate in 4 kbit/s units for the service to operate. This number is expected to be different from zero only for real-time traffic requiring a minimum transmission delay |
| BER | Service-level BER. Used in Rate Negotiation to select the desired PE that achieves the highest raw-bit rate and also meets BER requirements |
| Payload Encoding | Payload encoding used on logical channel<br>This parameter shall only be set when communicating flow parameters to the master. Between flow endpoints, the Payload Encoding is negotiated using Rate Negotiation. |

**Table 9-1/G.9954 – Flow properties**

| Field name | Description |
|---|---|
| Packet Timeout | The amount of time in ms a packet will remain queued before being deleted from the flow queue. A value of 0 indicates that packets never timeout and remain queued until transmitted on the line. |
| TX Timeslot | Timeslot of first TXOP defined for the flow. This field can be set by upper layers during flow setup in order to synchronize allocated TXOPs with an external source. This is intended for isochronous services. Time is measured in units of $2^{-13}$ ms with reference to the master's time reference as advertised in the Timestamp Report Indication; see 10.18. |
| Flow Inactivity Timeout | Amount of time (in ms) a flow will remain "alive", in the absence of any traffic, before the flow is automatically torn down and resources released. A value of 0 indicates that the flow is not automatically torn down. For further information on Flow Teardown, see 10.17. |

### 9.2.1 Source and destination address

The source and destination address of a flow is identified by the respective source and destination device addresses. The source address is a unicast 48-bit MAC address identifying the device at the source of the flow. The destination address identifies the destination of the flow and may be a unicast, multicast or broadcast 48-bit MAC address.

### 9.2.2 Flow ID

The Flow ID is a unique *flow identifier* between source and destination addresses. The Flow ID shall be assigned locally by the device at the source of the flow.

For more information on Flow IDs, see 7.3.2.

### 9.2.3 Service Class

A *Service Class* defines a collection of data flow (service) properties organized into a named class that can be simply identified (by enumerator) by upper layers entities.

This facility allows service properties to be globally defined and centrally maintained in the master. Upper protocol layers or endpoint nodes can provision service flows, by identifying them by enumerator without actually specifying the flow's QoS parameters. This implicitly defines a set of QoS parameters. In addition, amendments to the base QoS properties of a service class may be made by redefining individual flow parameters.

For a full list of the predefined service classes and their QoS parameters, see 7.3.2.

### 9.2.4 Priority classification

The priority classification represents the Link Layer priority assigned to the flow. The priority value has G.9951/2 semantics and is used to determine the PHY priority to use for transmissions when AMAC mode is used or when transmitting in a CTXOP (UTXOP). It may also be used by the scheduler in the master to rank flows in scheduling decisions.

Priority assignment should follow IEEE 802.1D and 802.1P recommendations for the mapping of user-priorities to traffic classes. For further information, see 10.17.

### 9.2.5 Service Type

The *Service Type* of a flow defines the type of QoS commitment guarantees required by the service. The following *Service Types* are defined:

**Table 9-2/G.9954 – Service types**

| Service Type | Description |
|---|---|
| Unsolicited Grant (CBR) | Supports real-time low-latency, fixed size, periodic (CBR) data. The resource scheduler guarantees allocation of a fixed amount of bandwidth periodically without explicit bandwidth requests. Used for "deterministic" QoS guarantees |
| Real-Time (rt-VBR) | Supports variable bit rate (VBR) data by supporting periodic variable size data grants. Suitable for MPEG video streams |
| Non Real-Time (nrt-VBR) | Similar to a real-time service except that the scheduler services Non Real-Time flows at a lower rate than real-time flows |
| Best Effort (BE) | Similar to non-real time service except that contention-based access may be defined within the allocated TXOP. NO guarantees (i.e., Best-Effort) are provided as to the frequency or length of TXOPs provided by the scheduler for Best-Effort services |

The *service type* parameter may be used by the master scheduler in scheduling decisions and by the source node in resource management decisions.

### 9.2.6 Maximum latency

This parameter defines the maximum tolerable transmission and queuing delay for a service. The parameter is defined by an enumerated value from a set of defined latencies expressed in ms.

The amount of latency a service can tolerate affects the amount of memory (buffer-space) required. For devices that have less buffer-space available than the amount implied by the latency parameter, an alternative (lesser) latency value may be specified by the destination device in the Flow Setup/Modify Response message used in the Flow Signalling Protocol.

The *Maximum Latency* parameter shall be used by the master scheduler in scheduling decisions concerning the length of TXOPs and the number of TXOPs assigned to the service within the MAC cycle. This parameter may also be used in AMAC mode to control the length of a burst of aggregated packets belonging to the same service.

For further details of the supported latency values and the Flow Signalling Protocol, see 10.17.

### 9.2.7 Maximum jitter

The maximum jitter parameter defines the maximum delay variation in latency values for a service above and below the mean latency value. Maximum jitter is expressed as (±Max) ms.

The *Maximum Jitter* parameter should be used by the master scheduler in scheduling decisions concerning the position of TXOPs within the MAC cycle.

Jitter values are expressed as an enumerated value within a set of defined jitter values. For further details of the supported jitter values, see the description in 10.17.

### 9.2.8 ACK policy

This flag indicates whether the flow requires link-layer acknowledgements using the LARQ mechanism in order to reduce the packet error rate (PER). ARQ is specified per ARQ channel where an ARQ channel is defined by a flow, i.e., by the tuple (*Source Address*, *Destination Address*, *Flow ID*) when operating in synchronous mode and by the (*Source Address*, *Destination Address*, *Priority*) when operating in AMAC mode.

NOTE – TCP-based protocols are natural candidates for applying a link-layer ACK policy as TCP performance may degrade significantly with an increase in packet errors.

### 9.2.9 Forward Error Control (FEC) policy

This flag indicates whether Reed-Solomon coding should be applied on the communication channel defined by the flow. This indication shall be used by a receiver to determine whether Reed-Solomon redundancy information should be sent to the transmitter, at the flow source, during Rate Negotiation.

### 9.2.10 Aggregation policy

Latency characteristics of a flow are used by the scheduler to determine how much flow data can be aggregated into a single transmission burst (frame). Scheduling decisions that account for a flow's latency requirements are taken by the master when calculating the size of a TXOP in the MAP. Similarly an endpoint device, performing local traffic scheduling (as in the case of AMAC mode of operation) may use latency characteristics to determine the amount of aggregation and transmission burst size.

Aggregation can be disabled completely for a flow, irrespective of the latency parameter, by specifying an aggregation policy of "*No Aggregation*".

NOTE – A "No Aggregation" policy may be useful when aggregation is performed at upper protocol layers and no further aggregation is desired.

### 9.2.11 CRC error handling policy

This clause specifies the policy to be used by the MAC when handling packets with CRC errors. Erroneous packets may be discarded by the MAC/link layers or passed up to higher protocol layers with erroneous bits in contained within.

The particular CRC error handling policy effects the semantics of the BER parameter as described in 9.2.14.

NOTE – Some services are tolerant of a small number of erroneous bits in the data stream. If the CRC Error Handling Policy specifies that erroneous packets should be discarded, then this implies a BER = 0 since no bit errors will be passed up to higher protocol layers. However, discarding complete packets introduces Packet Errors and the PER parameter becomes the dominant measure. A PER = 0 can be achieved using LARQ (ACK Policy) at the expense of latency.

### 9.2.12 Nominal packet size

The nominal packet size, in octets, for packets associated with the service. A value of 0 indicates an unspecified or unknown value.

### 9.2.13 Maximum, average and minimum data rates

The peak, average and minimum bit rates required for a service to operate effectively. Data rates are expressed in units of 4 kbit/s.

For CBR flows, the Minimum, Maximum and Average data rates are all equal. The minimum data rate is expected to be non-zero only for real-time traffic requiring a minimum transmission delay.

Given a service's Nominal Packet Size and Data Rates, it is possible to police and shape traffic into a form that conforms to the service specifications. This may be required in some implementations to ensure that a flow does not consume more resources than defined by its traffic specification. The allocation of TXOPs in the MAP inherently imposes traffic shaping on the endpoints.

### 9.2.14 Bit Error Ratio (BER)

Each service has an associated BER requirement that specifies the ratio of bit errors to "non-error" bits that a service is able to tolerate before QoS is affected.

The BER parameter is used to describe either the per-bit error probability, if packets with CRC errors delivered to upper layers, or the Packet Error Rate (PER) divided by the mean number of bits per packet, if packets with CRC errors are discarded. The policy for handling packets with CRC errors is specified by the CRC Error Handling Policy flag (see 9.2.11).

For example, consider a service using-1500 byte packets and requiring a PER = $10^{-2}$; then, $BER = 10^{-2}/(1500 \times 8) \approx 10^{-6}$.

NOTE – The service-level BER is used during Rate Negotiation to determine the best Payload Encoding that can be used in order to provide the highest throughput communication channel that is able to meet the BER requirements for a service. For further information on Rate Negotiation, see 10.4.

### 9.2.15 Payload encoding

This parameter defines the Payload Encoding (PE) to be used on the channel. The PE chosen is determined through Rate Negotiation and represents the PE providing the highest raw bit-rate that meets the BER parameter for the service.

### 9.2.16 TX timeslot

In order to support the synchronization of a flow's TXOPs with an external source (e.g., upstream timeslots in a broadband access network), the initiator of a flow setup sequence can indicate the timing of TXOPs desired on the home network. Timing is specified by an absolute time measured with respect to the master's time reference.

NOTE 1 – This feature requires an endpoint node to synchronize its clock with the master's clock reference using master Timestamp Reference Protocol. The time specified is an absolute time (remember we are synchronized with the master clock). The master knows the requested time and the max latency and so it can calculate where it should allocate the TXOPs in time. This parameter is only intended to help masters make scheduling decisions.

When allocating bandwidth for the specified flow, the master scheduler may use this information to influence the location of the associated TXOPs within the MAC cycle. When no timing information is provided, the master scheduler is free to allocate TXOPs as it sees fit. There is no requirement that the master meet the requested timing specification.

NOTE 2 – Timing information about the location of TXOPs is returned to upper convergence layers through the MAP mechanism. This allows upper layers to similarly synchronize to actual timing on the home network if so required.

For further details on master Clock Reference Synchronization, see 10.18.

### 9.2.17 Flow inactivity timeout

This parameter specifies the amount of time a flow may remain inactive before the flow is automatically "torndown" and its resources released. A flow is inactive in the absence of any traffic on the flow. A flow may be torn down by either device found at the endpoints of a flow.

A Flow Inactivity Timeout with a value of zero disables flow inactivity ageing.

NOTE – It is strongly suggested that flows be defined with inactivity ageing enabled in order to guarantee the release of media (and other) resources in case of service termination.

## 9.3 Convergence layer traffic classification

Packets from upper protocol layers are mapped to an underlying G.9954 flow by the Protocol Convergence Layer. The result of the mapping is a reference to the *flow descriptor* that describes the properties of the flow to which a packet belongs. The default mapping of a packet is to the *default flow* which defines contention-based access within UTXOPs on a priority-basis.

Packets are mapped to flows using *Traffic Classifiers*. A *Traffic Classifier* defines a protocol specific set of selection criteria that are applied to incoming packets in order to test their association with a specific flow. Several classifiers may be defined for a single flow and multiple classifiers

may be active at any one time in a Convergence Layer. Traffic classifiers are processed in an order that is defined by their relative priority.

Traffic classifiers may be installed in the Convergence Layer at the flow source by upper-layer management operations, during Network Admission or through Flow Setup/Modification Signalling operations.

For further details on the establishment of Convergence Layer Traffic Classification Filters, see the description of the Network Admission and Flow Signalling Protocols in clause 10.

### 9.4 Flow Signalling Protocol

To establish a flow with well-defined QoS parameters, as defined in 9.2, a *flow* shall be "set up" between source and destination devices. *Flow Setup* may be initiated by either the flow source or destination devices.

If a flow requires QoS guarantees, bandwidth for the flow shall be allocated in the MAP. To allocate bandwidth for a flow in the MAP, the master shall be informed of the setup of the flow.

*Flow Setup* shall be performed using the Flow Signalling Protocol and will involve a message exchange sequence, between the initiating node and the target node, whereby the initiator shall specify the properties of the flow (as defined in Table 9-1) to be set up.

To set up a flow with QoS Contracts, the device at the source of the flow shall notify the master using the same *Flow Signalling Protocol*. The master shall perform admission control on the flow setup request in order to determine whether sufficient media resources exist. If admitted, the master scheduler shall open TXOPs in the MAP that meet the QoS requirements of the requested flow. If the flow is not admitted, the master shall signal the error to the source of the flow setup request. It is implementation dependent as to the behaviour of a device upon failure to set up a flow.

NOTE – An implementation may tear down a flow if bandwidth cannot be reserved for it by the master. Alternatively, an implementation may continue to transmit data over the flow channel although fixed bandwidth cannot be reserved and other QoS parameters cannot be guaranteed.

During the lifetime of a flow, its specification may need to be modified in order to accommodate for changing (variable) bit-rate requirements, resource constraints (e.g., latency/jitter buffers) and achievable payload rates. Modifications to a flow specification are signalled between devices at the flow endpoints. In addition, if modifications to a flow's properties are such that they may affect media resource allocation in the MAP, the master shall be signalled, by the device at the source of the flow. Signalling shall be performed using the *Flow Modification* protocol. The master shall perform admission control on the requested flow modification.

During a Flow Modification, the QoS parameters that affect media resource allocation in the MAP are defined as follows:

Maximum, Average, Minimum Data Rates (see 9.2.13) – Change as a consequence of traffic statistics collected at flow source.

Payload Encoding (see 9.2.15) – Caused by changing line conditions and detected by Rate Negotiation.

Maximum Latency or Jitter (see 9.2.6 and 9.2.7) – Caused by changes in memory resource constraints at source or destination of flow.

Nominal Packet Size (see 9.2.12) – Caused by variableness in nature of packets in traffic stream.

Other flow properties are static and do not change during the lifetime of a flow.

When a flow is no longer needed or in use, it shall be torn down. *Flow Teardown* shall be performed by the Convergence Layer either explicitly, in response to a "teardown" request from upper layers, or implicitly through the aging out of inactive flows. If a flow has media resources

allocated to it (i.e., TXOPs in MAP), the master shall be signalled of the flow teardown by the device at the source of the flow. The master shall be informed using the *Flow Teardown* protocol. When a flow is torn down, the resources it binds shall be freed.

For a full description of the Flow Signalling Protocol, see 10.17.3.

## 9.5 Admission control

Admission control shall be performed by the master when a request is received to add a new flow or to change the properties of an existing flow to more stringent QoS parameters.

Admission control involves the following functions:

1) Bandwidth Testing;

2) Latency/Jitter Bound Testing.

Upon receiving a *Flow Setup* or *Modify* request, the master shall check for the availability of sufficient media resources (i.e., unallocated media time) in order to meet the flows throughput demands given the flows' *Minimum, Maximum* and *Average Data Rate* requirements and given the *Payload Encoding* required on the channel. Furthermore, the master shall verify that the location of available TXOPs is such that allocation of TXOPs to the flow will allow the flow to meet its requirements for Latency and Jitter bounds.

If admission control testing results in the failure of either or both of the admission control tests, the master shall return an ERROR in the *Flow Signalling "Response"* frame.

A flow's Latency/Jitter specification represents a maximum allowable bound and consequently the master may allocate media resources (TXOPs) in a manner such that it exceeds the original latency and jitter specification for the flow.

In order to meet QoS constraints of a flow specification, the master may need to reorganize the location and size of allocated TXOPs for other flows. This may be needed in order to "make space" for the addition of the new flow. The master should attempt to accommodate flows within the existing available media time before considering reorganization of other flows in order to localize the effect of the change in the Media Access Plan and in order NOT to introduce unnecessary (albeit transient) latency and jitter in other flows.

If admission control testing succeeds and the requested flow can be set up or modified according to the specified parameters, the master shall reserve media resources for the flow and advertise the reservation in the MAP.

For further information on the Flow Signalling Protocol, see 10.17.

## 9.6 QoS support in AMAC mode

When operating in G.9954 AMAC mode, QoS is supported using a combination of G.9951/2-style priority-based QoS together with the service-level flow specifications as described in 9.2.

In AMAC Mode, G.9954 nodes may set up, modify and teardown flows in a manner similar to operation in SMAC mode using the *Flow Signalling Protocol.* Flow signalling shall be performed between source and destination devices only.

Information contained in the flow specification may be used to support the asynchronous mode data transmissions. The fields of the flow specification that are relevant in AMAC mode and their semantics are defined in 9.2 and its subclauses.

NOTE – The use of flow specifications in AMAC mode allows a finer degree of control over the parameters used for the transmission of packets on the media. There is, however, no requirement that flows be explicitly managed in AMAC mode. Aggregation, priority selection, etc., may still be performed under the direction of upper protocol and convergence layers. This is an implementation decision.

# 10 Link Layer Protocol Specification

## 10.1 Overview

This Recommendation specifies the link layer format to be used for G.9954 stations. In addition, for link layer frames that are identified by IEEE assigned Ethertype value (0x886c) in the Type/Length field of the frame, these frames carry link control functionality, and definitions for this functionality are provided in this Recommendation.

The LLC sub-layer is responsible for performing link control functions. In particular, it is responsible for managing information concerning network connections, for enforcing Class of Service (CoS) and Quality of Service (QoS) constraints defined for the various service flows and for ensuring robust data transmission using Rate Negotiation, optional Reed-Solomon coding techniques and ARQ (Automatic Repeat ReQuest) techniques.

The following link control functions are defined in G.9954 Link Layer:

- Rate Negotiation;
- Link Integrity;
- Capability Announcement;
- Limited Automatic Repeat reQuest (LARQ);
- Frame bursting capability;
- MAC Cycle Synchronization;
- Registration;
- Flow Signalling;
- Master Selection;
- Certification Protocol;
- Collision Management Protocol;
- Reed Solomon Encapsulation;
- Timestamp Reporting.

These link functions use control frames to carry protocol messages between stations. G.9954 includes a standardized mechanism for Link Layer network control and encapsulation. Individual subtypes further distinguish control frames. The link control entities may be implemented in hardware or driver software. Link Control frames are not seen by layer 3 (IP) of the network stack, and shall not be bridged between network segments.

### 10.1.1 Minimal Link Protocol Support Profile for G.9954 Link Protocols

The Minimal Link Protocol Support Profile for G.9954 Link Protocols allows less complex implementations of this Recommendation. While all control protocols serve an important function in the operation of the network, it is possible to implement a minimal subset of Link Layer Protocols that are compatible with fully functional implementations and does not detract from the overall performance of other stations. The shorter name, Minimal Profile, will be used in the rest of this Recommendation.

Full support of all the link protocols, called the Full Link Protocol Support Profile, is assumed throughout the rest of this Recommendation unless Minimal Profile is explicitly mentioned.

A G.9954 device supporting a Minimal Profile shall support the following G.9954 Link Layer Protocols:

- Minimal Rate Negotiation;
- Link Integrity;
- Capability Announcement;
- MAC Cycle Synchronization;
- Frame Bursting;
- Certification Protocol;
- Minimal LARQ.

Such a device is able to synchronize with the master-generated synchronous MAC cycle and to contain its transmissions within Unallocated (Contention) TXOPs defined in the master-generated Media Access Plan (MAP). Media access is performed according to priority-based AMAC transmission rules. Frame bursting is used to more efficiently utilize media time. Rate Negotiation is performed over logical channels between source and destination devices.

### 10.1.2 G.9954 device supporting QoS contracts

In addition to the Link Layer Protocols in the Minimal Profile (above), a G.9954 device supporting QoS Contracts shall also support the following G.9954 Link Layer Protocols:

- Network Admission;
- Flow Signalling (Endpoint Device).

Such a device shall be able to perform all the functions of a Minimal Profile G.9954 device and shall also be able to manage *flows* with QoS Contracts, request bandwidth reservations for *flows* and perform Rate Negotiation and LARQ at the level (granularity) of a flow.

### 10.1.3 G.9954 master-capable device

A G.9954 device that is capable of becoming a network master, called a master-capable device for short, shall, in addition to the Link Layer Protocols described above also support the following G.9954 Link Layer Protocols:

- Dynamic Master Selection;
- MAC Cycle Generation;
- Flow Signalling (Master Device);
- Timestamp Reporting (Master Clock Reference).

A master-capable device shall be able to assume the role of master in a master-less network and to generate periodic MAC cycles for synchronous operation. It shall be able to engage in flow signalling and to convert flow signalling requests to scheduler input. A master-capable device shall be also able to act as a Master Clock Reference, by periodically advertising its internal clock allowing endpoint devices to synchronize their local clocks to the master's internal clock.

### 10.1.4 G.9954 optional Link Layer Protocols

The following Link Layer Protocols are optional for all G.9954 devices:

- Timestamp Reporting (Endpoint Slave);
- Collision Management Protocol;
- Reed Solomon Encapsulation.

## 10.2 Basic link layer frame format

The basic link layer frame format is described in Table 10-1.

**Table 10-1/G.9954 – Basic link layer format**

| Field | Length | Explanation |
|---|---|---|
| DA | 6 octets | Destination Address |
| SA | 6 octets | Source Address |
| Ethertype | 2 octets | Ethernet ethertype. Arbitrary value. If equal to **0x886c** (PNT Link Protocol Frame. Assigned by IEEE), then frame is for link protocol control frame. |
| Data | Variable | Payload data |
| Pad | Variable | Padding (if required to meet minimum length frame) |
| FCS | 4 octets | Frame Check Sequence |
| CRC-16 | 2 octets | PNT frame check sequence, described in 10.2.1 |

The G.9954 basic link layer frame format is based on the IEEE Std 802.3 Ethernet frame format (not including the IEEE Std 802.3 preamble and SFD fields) with an additional CRC-16 frame check sequence. The PNT frame bit fields starting with the destination address (DA) field and ending with the FCS field are identical to the corresponding fields described in IEEE Std 802.3 (see Figure 10-1), and are referred to as the Link-level Ethernet Frame. The bits of a PHY-level Ethernet frame have an Ethernet preamble and start-frame-delimiter (SFD) bits prepended to the Link-level frame; these bits are not present in G.9954 frames.

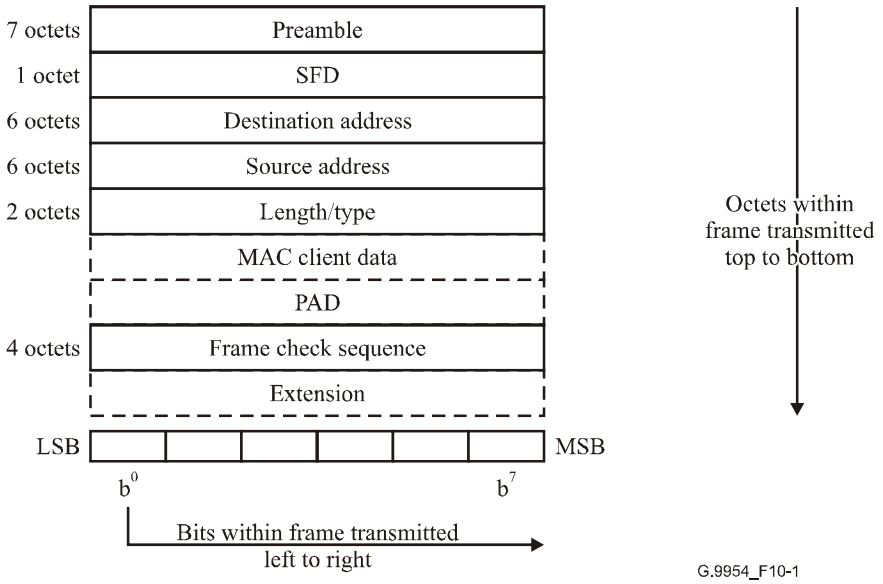


**Figure 10-1/G.9954 – Ethernet PHY level frame format**

It is intended that IEEE assigned Ethernet MAC addresses shall be used for Destination Address (DA) and Source Address (SA).

The Link Level Ethernet frame consists of an integer number of octets.

An additional CRC-16 shall be appended after the frame check sequence, as described in 10.2.1.

In the frame formats defined above, before transmission, the Link Control Frame shall be converted into a G.9954 physical layer frame by adding Preamble, Frame Control, PAD and EOF as shown in Figure 6-2.

### 10.2.1 CRC-16

A 16-bit cyclic redundancy check (CRC) shall be computed as a function of the contents of the (unscrambled) Ethernet Link-Level frame in transmission order, starting with the first bit of the DA field and ending with the last bit of the FCS field. The encoding is defined by the following generating polynomial:

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

Mathematically, the CRC value corresponding to a given frame is defined by the following procedure.

The first 16 bits of the frame in transmission order are complemented.

The $n$ bits of the frame in transmission order are then considered to be the coefficients of a polynomial $M(x)$ of degree $n - 1$. (The first bit of the Destination Address field corresponds to the $x^{(n-1)}$ term and the last bit of the FCS field corresponds to the $x^0$ term.)

$M(x)$ is multiplied by $x^{16}$ and divided by $G(x)$, producing a remainder $R(x)$ of degree $\leq 15$.

The coefficients of $R(x)$ are considered to be a 16-bit sequence.

The bit sequence is complemented and the result is the CRC.

The 16 bits of the CRC shall be placed in the CRC-16 field so that $x^{15}$ is the least significant bit of the first octet, and the $x^0$ term is the most-significant bit of the last octet. (The bits of the CRC are thus transmitted in the order $x^{15}$, $x^{14}$, .. $x^1$, $x^0$.)

NOTE – The PNT CRC-16, in conjunction with Ethernet's FCS, provides more protection from undetected errors than the FCS alone. This is motivated by environmental factors that will often result in a frame error rate (FER) several orders of magnitude higher than that of Ethernet, making the FCS insufficient by itself.

## 10.3 Link layer control frames

Link layer frames with ethertypes equal to **0x886c** are **link layer control frames.** These frames are not based on the IEEE Std 802.3 Ethernet frame format. There are two basic formats for a Link Control Frame, a long subtype and a short subtype. The long subtype format is provided for future specified control frames where the amount of control information exceeds 256 octets. The control and encapsulation frames described in this Recommendation use the short subtype format.

In the frame formats defined below, before transmission the Link Control Frame shall be converted into a physical layer frame by adding Preamble, Frame Control, PAD and EOF as shown in Figure 6-2.

### 10.3.1 Short format

The short-format link control frame is defined in Table 10-2. The SSVersion field should be used by all protocols using the Short-Format Link Control Frame header. This field specifies which format version of the control information is used. This allows future extension of each SSType.

**Table 10-2/G.9954 – Short-format link control frame**

| Field | Length | Explanation |
|---|---|---|
| DA | 6 octets | Destination Address |
| SA | 6 octets | Source Address |
| Ethertype | 2 octets | **0x886c** (PNT Link Protocol Frame Assigned by IEEE) |
| SSType | 1 octet | 0-127 assigned by PNT<br>0 Reserved<br>1 Rate Request Control Frame<br>2 Link Integrity Short Frame<br>3 Capabilities Announcement<br>4 LARQ<br>5 Vendor-specific short-format type<br>6 Frame bursting<br>7 Dynamic Master Selection<br>8 Timestamp Report Indication<br>9-127 Reserved<br>Values 128-255 correspond to the Long Subtype. |
| SSLength | 1 octet | Number of additional octets in the control header, starting with the SSVersion field (or the first octet following SSLength if it is not defined as SSVersion) and ending with the second (last) octet of the Next Ethertype field. Min is 2 and max is 255. |
| SSVersion | 1 octet | Version number of the control information |
| Control Data | 0-252 octets | Control information |
| Next Ethertype | 2 octets | Ethertype/length of next layer protocol, 0 if none. |
| Payload data | Variable | If not encapsulating frame, then this field is 0 octets long. |
| Pad | 41-0 octets | Padding required to meet minimum if data <41 octets |
| FCS | 4 octets | Frame Check Sequence |
| CRC-16 | 2 octets | PNT Frame Check Sequence |

SSLength must be checked to ensure that enough control information is present. New, backwards compatible, frame formats may contain additional fixed data fields, but shall always contain the fixed fields specified in earlier formats. Protocol implementations must interpret all supported SSType frames using the latest supported SSVersion that is less than or equal to the SSVersion indicated in the received frame. Unknown fields shall be ignored. Encapsulated data from unsupported (newer) SSVersions of supported encapsulating SSType frames shall be passed to the layer above. Protocol extensibility is addressed in 10.10.

The Next Ethertype field is required for all Short-Format Link Control Frame headers. Among other things, it supports backward compatibility by enabling receivers to always strip short-format link layer headers. If the Next Ethertype field is zero, then the frame is a basic control frame and should be dropped after processing the control information it contains. The Next Ethertype field shall be the last two octets of the control header. The position of the Next Ethertype field in the frame shall be determined using the SSLength field in order to ensure forward compatibility.

If the Next Ethertype field is non-zero, then the frame is an *encapsulating* control frame. An encapsulated data frame is an encapsulating control frame with any Next Ethertype field not matching x0000 or x886c. G.9954 receivers shall be capable of removing at least one encapsulating Short-Format Link Control Frame header from any received encapsulated data frame. When Next

Ethertype is restricted by the specification to the value x0000 for a specific Link Layer control frame SSType or LSType, then encapsulation of data frames is not allowed when using that Link Layer control frame type. The only Link Layer frame type that supports encapsulation of data frames is the LARQ frame.

If the SSType is not understood by the receiver (a fact possibly announced via future CSA options), then the frame shall be dropped. All nodes are required to understand the LARQ SSType (although they are not required to implement LARQ). Protocol extensibility is addressed in 10.10.

The header and trailer for standard Ethernet frames are shaded with grey, in order to highlight the formats of the control information frames.

### 10.3.2 Long format

The long-format link control frame is defined in Table 10-3. An LSVersion, similar to SSVersion, should be used by all Long-Format subtypes. A Next_Ethertype field is required for all Long-Format subtypes. If Long-Format subtypes (LSType values) are not understood by the receiver (a fact possibly announced via future CSA options) then they shall be dropped. Processing requirements with respect to forwards compatibility, dropping of unknown frame types with Next_Ethertype = 0, and removal of Long-Format headers with Next_Ethertype ! = 0, are identical to those for Short-Format Control Frame headers.

**Table 10-3/G.9954 – Long-format link protocol frame**

| Field | Length | Explanation |
|---|---|---|
| DA | 6 octets | Destination Address |
| SA | 6 octets | Source Address |
| Ethertype | 2 octets | **0x886c** (PNT Link Protocol Frame.Assigned by IEEE) |
| LSType | 2 octets | 32768 Reserved<br>32769 Vendor-specific long format<br>32770 Certification protocol<br>32771 Reed-Solomon encapsulating header<br>32772 MAP Synchronization Protocol<br>32773 Network Admission Protocol<br>32774 Flow Signalling Protocol<br>32775 to 65534 Reserved, assigned by PNT<br>65535 Reserved |
| LSLength | 2 octets | Number of additional octets in the control header, starting with the SSVersion field (or the first octet following SSLength if it is not defined as SSVersion) and ending with the second (last) octet of the Next Ethertype field. Min is 2 and max is 65535. |
| LSVersion | 1 octet | Version number of the following protocol information |
| Data | LSLength – 3 octets | LSType protocoldependent data |
| Next Ethertype | 2 octets | Ethertype/length of next layer protocol, 0 if none. |
| Payload Data | Variable | If not encapsulating frame, then this field is 0 octets long. |
| Pad | 42-0 octets | Pad to minimum size if needed |
| FCS | 4 octets | Frame Check Sequence |
| CRC-16 | 2 octets | PNT Frame Check Sequence |

### 10.3.3 Order of transmission

Network transmission order of frame fields is from the top to the bottom of each table.

Within a field, the MSByte of the field shall be the first octet of the field to be transmitted, with the LSBit of each octet transmitted first. Subsequent bytes within a field are transmitted in decreasing order of significance.

When subfields are indicated in any Table, the ordering shown is decreasing significance from the top to the bottom of the Table.

## 10.4 Rate Negotiation control function

The PHY payload modulation can use 2 to 8 bits-per-symbol constellations and one of several defined *bands* which are combinations of bauds, modulation type and spectral mask. For some bands 8, 9 and 10 bits-per-symbol constellations optionally exist; see 6.3.3.4.

The payload encoding (PE) that can be achieved is a function of the channel quality between source and destination, and the channel quality generally differs between each pair of stations depending on the wiring topology and specific channel impairments. Therefore the Rate Negotiation function in a destination station uses Rate Request Control Frames (RRCF) to provide information to a source station as to the payload encoding that the source station should use to encode future frames sent to this destination, and to generate test frames to assist a receiver in selecting the most appropriate band to use.

The policy that the destination station uses to select the desired payload encoding and the policy it uses to decide when to transmit Rate Request Control Frames are implementation dependent. Stations should avoid transmission policies that can result in excessive RRCF traffic.

Rate Negotiation in this Recommendation is similar to that defined in ITU-T Rec. G.9951/2 except that the meaning of a logical channel (see Terms and Definitions in 10.4.3.1) in G.9954 is extended to include the logical channel defined by the tuples { Source Address, Destination Address, Priority } and { Source Address, Destination Address, Flow ID }. This extension allows a fine degree of control over the selected rate for a logical channel by allowing different rates to be negotiated per logical channel, even when the different channels are over the same Source-Destination pair. Since each logical channel represents a different service or flow, possibly with distinct BER/PER requirements, Rate Negotiation is adaptive per-service.

The goal of Rate Negotiation is to select the Payload Encoding that achieves the highest raw bit rate while still meeting the BER/PER requirements for the logical channel.

### 10.4.1 Rate Request Control Frame format

The RRCF specifies a maximum constellation (bits per symbol) that the receiver (ReqDA) wishes to be used in a given band, or indicates that a given band is not supported. (See Table 10-4.)

**Table 10-4/G.9954 – Rate request control frame definition**

| Field | Length | Meaning |
|---|---|---|
| DA | 6 octets | Destination Address |
| SA | 6 octets | Source Address |
| Ethertype | 2 octets | 0x886c (PNT Link Control Frame) |
| SSType | 1 octet | = SUBTYPE_RATE (1) |
| SSLength | 1 octet | Number of additional octets in the control header, starting with the SSVersion field and ending with the second (last) octet of the Next Ethertype field. The minimum value of SSLength is 18 for SSVersion 0. |

**Table 10-4/G.9954 – Rate request control frame definition**

| Field | Length | Meaning |
|---|---|---|
| SSVersion | 1 octet | = 0 |
| OpCode | 1 octet | Operation code for this control message. See Table 10-6 for definitions. |
| NumBands | 1 octet | Number of bands specified in this control. Each band has a two octet descriptor. The bands refer to the modulation type:<br>*Band* *Reference*<br>1 Spectral Mask #1, 2-Mbaud modulation<br>2 Spectral Mask #1, 4-Mbaud modulation<br>3 Spectral Mask #2, 2-Mbaud modulation<br>4 Spectral Mask #2, 4-Mbaud modulation<br>5 Spectral Mask #2, 8-Mbaud modulation<br>6 Spectral Mask #2, 16-Mbaud modulation<br>7 Spectral Mask #3, 2-Mbaud modulation<br>8 Spectral Mask #3, 6-Mbaud modulation<br>9 Spectral Mask #3, 12-Mbaud modulation<br>10 Spectral Mask #3, 24-Mbaud modulation<br>NumBands shall be 6 or 10 on transmission for G.9954 stations, and stations shall ignore band entries beyond Band10 on receive if NumBands is larger than 10. The value 0 is not allowed. Values greater than 6 can be ignored if the G.9954 station does not support Spectral Mask #3. |
| NumAddr | 1 octet | Number of addresses specified in the payload of this control message. NumAddr may be zero. The SA in the Ethernet header is always used, and is referred to in the following clauses as RefAddr0. |
| Band1_PE | 1 octet | The PE value that should be used to send data when Band 1 is selected. |
| Band1_rank | 1 octet | The rank order of the ReqDAs' preference for this Band, 1 is highest preference, and the other bands within the spectral mask are assigned successively larger rank values. |
| • • • | | Additional instances of Band information |
| BandN_PE | 1 octet | The PE value that should be used to send data when Band N is selected |
| BandN_rank | 1 octet | The rank order of the ReqDAs' preference for this Band, 1 is highest preference, and the other bands within the spectral mask are assigned successively larger rank values. |
| RefAddr1 | 6 octets | Optional. Present if NumAddr ≥ 1. The second MAC address for which the rates are being specified, only broadcast and multicast address types are allowed. |
| RefAddr2 | 6 octets | Optional. Present if NumAddr ≥ 2. The third MAC address for which the rates are being specified. Only broadcast and multicast address types are allowed. |
| • • • | | [additional instances of RefAddr, until the number of RefAddr fields equals NumAddr] |

**Table 10-4/G.9954 – Rate request control frame definition**

| Field | Length | Meaning |
|---|---|---|
| [Additional TLV extensions] | | Flow ID/Priority extension information. See 10.4.2. |
| Next Ethertype | 2 octets | = 0 |
| Pad | | To reach minFrameSize if required |
| FCS | 4 octets | Frame Check Sequence |
| CRC-16 | 2 octets | PNT Frame Check Sequence |

Additional bands may exist in future versions of this Recommendation, and can be described with band descriptors {PE, rank} added after Band 10. If additional bands are present, their descriptors will appear between BandN_Rank and RefAddr1, and G.9954 stations take their presence into account when determining the location of the RefAddr list.

G.9954 stations shall ignore band specification beyond Numbands = 10. If a receiver does not specify a band in an RRCF, or specifies a PE of 0 for a band, then transmitters shall not use that band. In order to allow unambiguous determination of which bands are present as future bands are added, intervening unsupported bands must use PE = 0 to indicate non-use. Bands may only be unspecified if no other band information follows.

The NumBands and NumAddr fields are placed next to each other so that all the fixed fields can be referenced at known offsets in the frame.

Table 10-5 describes the assigned values that may appear in the band description entries in the Rate Request Control Frame.

**Table 10-5/G.9954 – PE values for rate request control frames**

| PE | Data rate | Meaning |
|---|---|---|
| 0 | N/A | Means this band is not supported |
| 1 | 4 Mbit/s | Spectral Mask #1, 2-Mbaud FDQAM, 2 bits per symbol |
| 2 | 6 Mbit/s | Spectral Mask #1, 2-Mbaud FDQAM, 3 bits per symbol |
| 3 | 8 Mbit/s | Spectral Mask #1, 2-Mbaud FDQAM, 4 bits per symbol |
| 4 | 10 Mbit/s | Spectral Mask #1, 2-Mbaud FDQAM, 5 bits per symbol |
| 5 | 12 Mbit/s | Spectral Mask #1, 2-Mbaud FDQAM, 6 bits per symbol |
| 6 | 14 Mbit/s | Spectral Mask #1, 2-Mbaud FDQAM, 7 bits per symbol |
| 7 | 16 Mbit/s | Spectral Mask #1, 2-Mbaud FDQAM, 8 bits per symbol |
| 8 | 1 Mbit/s | Reserved for legacy systems |
| 9 | 8 Mbit/s | Spectral Mask #1, 4-Mbaud QAM, 2 bits per symbol |
| 10 | 12 Mbit/s | Spectral Mask #1, 4-Mbaud QAM, 3 bits per symbol |
| 11 | 16 Mbit/s | Spectral Mask #1, 4-Mbaud QAM, 4 bits per symbol |
| 12 | 20 Mbit/s | Spectral Mask #1, 4-Mbaud QAM, 5 bits per symbol |
| 13 | 24 Mbit/s | Spectral Mask #1, 4-Mbaud QAM, 6 bits per symbol |
| 14 | 28 Mbit/s | Spectral Mask #1, 4-Mbaud QAM, 7 bits per symbol |
| 15 | 32 Mbit/s | Spectral Mask #1, 4-Mbaud QAM, 8 bits per symbol |
| 16-32 | N/A | Reserved |

**Table 10-5/G.9954 – PE values for rate request control frames**

| PE | Data rate | Meaning |
|---|---|---|
| 33 | 4 Mbit/s | Spectral Mask #2, 2-Mbaud FDQAM, 2 bits per symbol |
| 34 | 6 Mbit/s | Spectral Mask #2, 2-Mbaud FDQAM, 3 bits per symbol |
| 35 | 8 Mbit/s | Spectral Mask #2, 2-Mbaud FDQAM, 4 bits per symbol |
| 36 | 10 Mbit/s | Spectral Mask #2, 2-Mbaud FDQAM, 5 bits per symbol |
| 37 | 12 Mbit/s | Spectral Mask #2, 2-Mbaud FDQAM, 6 bits per symbol |
| 38 | 14 Mbit/s | Spectral Mask #2, 2-Mbaud FDQAM, 7 bits per symbol |
| 39 | 32 Mbit/s | Spectral Mask #2, 2-Mbaud FDQAM, 8 bits per symbol |
| 40 | N/A | Reserved |
| 41 | 8 Mbit/s | Spectral Mask #2, 4-Mbaud FDQAM, 2 bits per symbol |
| 42 | 12 Mbit/s | Spectral Mask #2, 4-Mbaud FDQAM, 3 bits per symbol |
| 43 | 16 Mbit/s | Spectral Mask #2, 4-Mbaud FDQAM, 4 bits per symbol |
| 44 | 20 Mbit/s | Spectral Mask #2, 4-Mbaud FDQAM, 5 bits per symbol |
| 45 | 24 Mbit/s | Spectral Mask #2, 4-Mbaud FDQAM, 6 bits per symbol |
| 46 | 28 Mbit/s | Spectral Mask #2, 4-Mbaud FDQAM, 7 bits per symbol |
| 47 | 32 Mbit/s | Spectral Mask #2, 4-Mbaud FDQAM, 8 bits per symbol |
| 48 | N/A | Reserved |
| 49 | 16 Mbit/s | Spectral Mask #2, 8-Mbaud FDQAM, 2 bits per symbol |
| 50 | 24 Mbit/s | Spectral Mask #2, 8-Mbaud FDQAM, 3 bits per symbol |
| 51 | 32 Mbit/s | Spectral Mask #2, 8-Mbaud FDQAM, 4 bits per symbol |
| 52 | 40 Mbit/s | Spectral Mask #2, 8-Mbaud FDQAM, 5 bits per symbol |
| 53 | 48 Mbit/s | Spectral Mask #2, 8-Mbaud FDQAM, 6 bits per symbol |
| 54 | 56 Mbit/s | Spectral Mask #2, 8-Mbaud FDQAM, 7 bits per symbol |
| 55 | 64 Mbit/s | Spectral Mask #2, 8-Mbaud FDQAM, 8 bits per symbol |
| 56 | N/A | Reserved |
| 57 | 32 Mbit/s | Spectral Mask #2, 16-Mbaud QAM, 2 bits per symbol |
| 58 | 48 Mbit/s | Spectral Mask #2, 16-Mbaud QAM, 3 bits per symbol |
| 59 | 64 Mbit/s | Spectral Mask #2, 16-Mbaud QAM, 4 bits per symbol |
| 60 | 80 Mbit/s | Spectral Mask #2, 16-Mbaud QAM, 5 bits per symbol |
| 61 | 96 Mbit/s | Spectral Mask #2, 16-Mbaud QAM, 6 bits per symbol |
| 62 | 112 Mbit/s | Spectral Mask #2, 16-Mbaud QAM, 7 bits per symbol |
| 63 | 128 Mbit/s | Spectral Mask #2, 16 Mbaud, QAM, 8 bits per symbol |
| 64 | | Reserved |
| 65 | 4 Mbit/s | Spectral Mask #3, 2-Mbaud FDQAM, 2 bits per symbol |
| 66 | 6 Mbit/s | Spectral Mask #3, 2-Mbaud FDQAM, 3 bits per symbol |
| 67 | 8 Mbit/s | Spectral Mask #3, 2-Mbaud FDQAM, 4 bits per symbol |
| 68 | 10 Mbit/s | Spectral Mask #3, 2-Mbaud FDQAM, 5 bits per symbol |
| 69 | 12 Mbit/s | Spectral Mask #3, 2-Mbaud FDQAM, 6 bits per symbol |
| 70 | 14 Mbit/s | Spectral Mask #3, 2-Mbaud FDQAM, 7 bits per symbol |

**Table 10-5/G.9954 – PE values for rate request control frames**

| PE | Data rate | Meaning |
|---|---|---|
| 71 | 16 Mbit/s | Spectral Mask #3, 2-Mbaud FDQAM, 8 bits per symbol |
| 72 | N/A | Reserved |
| 73 | 12 Mbit/s | Spectral Mask #3, 6-Mbaud FDQAM, 2 bits per symbol |
| 74 | 18 Mbit/s | Spectral Mask #3, 6-Mbaud FDQAM, 3 bits per symbol |
| 75 | 24 Mbit/s | Spectral Mask #3, 6-Mbaud FDQAM, 4 bits per symbol |
| 76 | 30 Mbit/s | Spectral Mask #3, 6-Mbaud FDQAM, 5 bits per symbol |
| 77 | 36 Mbit/s | Spectral Mask #3, 6-Mbaud FDQAM, 6 bits per symbol |
| 78 | 42 Mbit/s | Spectral Mask #3, 6-Mbaud FDQAM, 7 bits per symbol |
| 79 | 48 Mbit/s | Spectral Mask #3, 6-Mbaud FDQAM, 8 bits per symbol |
| 80 | N/A | Reserved |
| 81 | 24 Mbit/s | Spectral Mask #3, 12-Mbaud FDQAM, 2 bits per symbol |
| 82 | 36 Mbit/s | Spectral Mask #3, 12-Mbaud FDQAM, 3 bits per symbol |
| 83 | 48 Mbit/s | Spectral Mask #3,12-Mbaud FDQAM, 4 bits per symbol |
| 84 | 60 Mbit/s | Spectral Mask #3, 12-Mbaud FDQAM, 5 bits per symbol |
| 85 | 72 Mbit/s | Spectral Mask #3, 12-Mbaud FDQAM, 6 bits per symbol |
| 86 | 84 Mbit/s | Spectral Mask #3, 12-Mbaud FDQAM, 7 bits per symbol |
| 87 | 96 Mbit/s | Spectral Mask #3, 12-Mbaud FDQAM, 8 bits per symbol |
| 88 | N/A | Reserved |
| 89 | 48 Mbit/s | Spectral Mask #3, 24-Mbaud FDQAM, 2 bits per symbol |
| 90 | 72 Mbit/s | Spectral Mask #3, 24-Mbaud FDQAM, 3 bits per symbol |
| 91 | 96 Mbit/s | Spectral Mask #3, 24-Mbaud FDQAM, 4 bits per symbol |
| 92 | 120 Mbit/s | Spectral Mask #3, 24-Mbaud FDQAM, 5 bits per symbol |
| 93 | 144 Mbit/s | Spectral Mask #3, 24-Mbaud FDQAM, 6 bits per symbol |
| 94 | 168 Mbit/s | Spectral Mask #3, 24-Mbaud FDQAM, 7 bits per symbol |
| 95 | 192 Mbit/s | Spectral Mask #3, 24-Mbaud FDQAM, 8 bits per symbol |
| 96-159 | N/A | Reserved |
| 160 | 16 Mbit/s | Spectral Mask #2, 2-Mbaud FDQAM, 8-round constellation; 8 bits per symbol |
| 161 | 18 Mbit/s | Spectral Mask #2, 2-Mbaud FDQAM, 9-round constellation; 9 bits per symbol |
| 162 | 20 Mbit/s | Spectral Mask #2, 2-Mbaud FDQAM, 10-round constellation; 10 bits per symbol |
| 163-167 | N/A | Reserved |
| 168 | 32 Mbit/s | Spectral Mask #2, 4-Mbaud FDQAM, 8-round constellation; 8 bits per symbol |
| 169 | 36 Mbit/s | Spectral Mask #2, 4-Mbaud FDQAM, 9-round constellation; 9 bits per symbol |
| 170 | 40 Mbit/s | Spectral Mask #2, 4-Mbaud FDQAM, 10-round constellation; 10 bits per symbol |
| 171-175 | N/A | Reserved |
| 176 | 64 Mbit/s | Spectral Mask #2, 8-Mbaud FDQAM, 8-round constellation; 8 bits per symbol |
| 177 | 72 Mbit/s | Spectral Mask #2, 8-Mbaud FDQAM, 9-round constellation; 9 bits per symbol |
| 178 | 80 Mbit/s | Spectral Mask #2, 8-Mbaud FDQAM, 10-round constellation; 10 bits per symbol |
| 179-183 | N/A | Reserved |

**Table 10-5/G.9954 – PE values for rate request control frames**

| PE | Data rate | Meaning |
|---|---|---|
| 184 | 128 Mbit/s | Spectral Mask #2, 16-Mbaud FDQAM, 8-round constellation; 8 bits per symbol |
| 185 | 144 Mbit/s | Spectral Mask #2, 16-Mbaud FDQAM, 9-round constellation; 9 bits per symbol |
| 186 | 160 Mbit/s | Spectral Mask #2, 16-Mbaud FDQAM, 10-round constellation; 10 bits per symbol |
| 187-191 | N/A | Reserved |
| 192 | 16 Mbit/s | Spectral Mask #3, 2-Mbaud FDQAM, 8-round constellation; 8 bits per symbol |
| 193 | 18 Mbit/s | Spectral Mask #3, 2-Mbaud FDQAM, 9-round constellation; 9 bits per symbol |
| 194 | 20 Mbit/s | Spectral Mask #3, 2-Mbaud FDQAM, 10-round constellation; 10 bits per symbol |
| 195-199 | N/A | Reserved |
| 200 | 48 Mbit/s | Spectral Mask #3, 6-Mbaud FDQAM, 8-round constellation; 8 bits per symbol |
| 201 | 54 Mbit/s | Spectral Mask #3, 6-Mbaud FDQAM, 9-round constellation; 9 bits per symbol |
| 202 | 60 Mbit/s | Spectral Mask #3, 6-Mbaud FDQAM, 10-round constellation; 10 bits per symbol |
| 203-207 | N/A | Reserved |
| 208 | 96 Mbit/s | Spectral Mask #3, 12-Mbaud FDQAM, 8-round constellation; 8 bits per symbol |
| 209 | 108 Mbit/s | Spectral Mask #3, 12-Mbaud FDQAM, 9-round constellation; 9 bits per symbol |
| 210 | 120 Mbit/s | Spectral Mask #3, 12-Mbaud FDQAM, 10-round constellation; 10 bits per symbol |
| 211-215 | N/A | Reserved |
| 216 | 192 Mbit/s | Spectral Mask #3, 24-Mbaud FDQAM, 8-round constellation; 8 bits per symbol |
| 217 | 216 Mbit/s | Spectral Mask #3, 24-Mbaud FDQAM, 9-round constellation; 9 bits per symbol |
| 218 | 240 Mbit/s | Spectral Mask #3, 24-Mbaud FDQAM, 10-round constellation; 10 bits per symbol |
| 219-255 | N/A | Reserved |

Table 10-6 describes the values that may appear in the OpCode entry in the Rate Request Control Frame.

**Table 10-6/G.9954 – OpCode values for rate request control frames**

| OpCode | Meaning |
|---|---|
| 0 | Rate Change Request |
| 1 | Rate Test Request |
| 2 | Rate Test Reply |
| 3-255 | Reserved |

### 10.4.2 Receiver indication of logical channel. TLV extension to the LCP SUBTYPE_RATE Subtype

In order to support Rate Negotiation over a logical channel defined by { Source Address, Destination Address, Priority } or { Source Address, Destination Address, Flow ID }, a TLV extension to the Rate Request Control Frame (RRCF) is defined.

Two additional parameters are included for each RefAddr defined in the RRCF. These parameters indicate the Priority or Flow Identifier for the logical channel whose Source Address is the DA in the Ethernet header of the RRCF frame and whose Destination Address = RefAddr<n>.

There are three types of logical channels defined for Rate Negotiation. These are as follows:

1) Simple Channel – equivalent to the G.9951/2 definition of logical channel and defined by the { Source Address, Destination Address } pair. No additional channel identifier is required. For a simple channel a PER=1e-4 shall be used as the PER parameter for Rate Selection.

2) LARQ Priority Channel – equivalent to the G.9951/2 definition of logical channel and defined by the tuple { Source Address, Destination Address, Priority }. For a LARQ priority channel, a PER=1e-2 shall be used as the PER parameter for Rate Selection.

3) Flow Channel – defines the logical channel identified by { Source Address, Destination Address, Flow ID }. The BER/PER used as input for Rate Selection is defined in the Flow Parameters negotiated and signalled between Source and Destination during flow signalling. For further information on Flow Parameters and the Flow Signalling Protocol, see 10.17.

The Logical Channel Identifier TLV extension is optional. If, however, the extension is found in the RRCF frame, there shall be a pair of parameters (RefChanType<n>, RefChanId<n>) for each RefAddr in the frame (i.e., NumAddr+1 entries). The first entry shall correspond to RefAddr0 and the last entry to $RefAddr_{NumAddr}$. (See Table 10-7.)

**Table 10-7/G.9954 – Flow ID/priority extension information**

| Field | Length | Meaning |
|---|---|---|
| SETag | 1 octet | = 3, Optional logical channel identifier |
| SELength | 1 octet | Total length of TLV extension excluding the tag and length octets<br>Must be (NumAddr+1) × 2. Minimum is 4 |
| RefChanType0 | 1 octet | The logical channel type defined by (DA, RefAddr0, RefId0). The channel type defines the semantics of RefId as follows:<br>0 Simple channel: RefId is undefined.<br>1 LARQ priority channel: RefId is interpreted as Priority.<br>2 Flow channel: RefId is interpreted as FlowId. |
| RefId0 | 1 octet | The RefId according to the semantics defined by RefChanType |
| RefChanType1 | 1 octet | The logical channel type defined by (DA, RefAddr1, RefId1). The channel type defines the semantics of RefId as follows:<br>0 Simple channel: RefId is undefined.<br>1 LARQ priority channel: RefId is interpreted as Priority.<br>2 Flow channel: RefId is interpreted as FlowId. |
| RefId1 | 1 octet | The RefId according to the semantics defined by RefChanType1 |
| • • • | | [additional instances of Channel Identification information, until the number of channels equals NumAddr+1. The Channel Identification Table is optional as indicated by TLV extension mechanism. If the TLV extension does not exist, all logical channels are assumed simple channels. Otherwise, there must be an explicit Channel Identification entry for each defined RefAddr from RefAddr0..$RefAddr_{NumAddr}$] |

### 10.4.3 Terms and definitions

**Table 10-8/G.9954 – Terms and definitions**

| Term | Definition |
|---|---|
| band specification | A Payload Encoding (PE) and Rank associated with a given band. A band is a single combination of baud, modulation type (e.g., QAM or FDQAM) and carrier frequency. Ten bands are defined in this Recommendation. |
| Logical channel, channel | A flow of frames from a sender to one or more receivers on a single network segment, consisting of all the frames with a single combination of 1) DA and SA, or 2) DA, SA and Priority, or 3) DA, SA and Flow ID. Each combination represents a different channel type referred to as a Simple, LARQ Priority and Flow Channels respectively. |
| Receiver | A station that receives frames sent on a particular channel. If the destination is a unicast address there is at most one receiver. If the destination is a group address (including broadcast), there may be many receivers. |
| Receiver PE | The preferred PE to be used on this channel, as determined by the receiver. |
| RRCF | Rate Request Control Frame. Sent from the receiver to the sender to effect a change in PE. |
| RefAddr0 | The SA in the Ethernet header of the RRCF frame. This is the DA of the receiver (for the channel), and is always used by the channel sender as the first RefAddr processed. |
| RefAddr1..RefAddr<n> | Other addresses including broadcast and multicast addresses for which the receiver is indicating rate information to the sender. The channel receiver's station address (RefAddr0) should not be put in the list of additional RefAddrs.<br>NOTE – At least one RefAddr field is necessary to support rate negotiation for broadcast and multicast addresses since these cannot be used as the source address in the Ethernet header. |
| Sender | The sending station for a channel, usually the station owning the source MAC address |
| Sender PE | The preferred PE associated with a channel, as noted by the sender |

#### 10.4.3.1 Channels

Rate Negotiation is defined over simplex logical channels. A separate channel is defined for each combination of Ethernet 1) DA, SA or 2) DA, SA and Priority or 3) DA, SA and Flow ID. The different combinations represent different channel types and are referred to as Simple, LARQ Priority and Flow Channels respectively. There is no explicit channel setup procedure for Simple and LARQ priority channels. A new channel is implicitly defined when a packet is received from a new SA or sent to a new DA. Flow Channels are set up by Flow Signalling between channel Source and Destination devices. For further information on the Flow Signalling Protocol, see 10.17.

Each channel has a single sender but can have multiple **receivers**. **Receivers** operate independently.

#### 10.4.3.2 Sending RRCFs

Rate control frames (all OpCodes) should be sent with a priority corresponding to Link Layer priority 7. RRCFs shall never be sent with a Link Layer priority of 6. RRCFs may be sent with a lower Link Layer priority, from the set [5,4,3,0]. However, the Link Layer priority of an RRCF shall never be lower than the highest Link Layer priority received in the last 2 seconds from the station to which the RRCF is being sent. Rate Change Requests (OpCode = 0) shall always be sent

with an encoding of Spectral Mask #2, 2-Mbaud FDQAM at 2 bits per symbol (PE = 33) when the channel source is a G.9954 device. Spectral Mask #1, 2-Mbaud FDQAM at 2 bits per symbol (PE = 1) shall be used when the channel source device is a G.9954 device operating G.9951/2 mode. Selection of the encoding for Rate Test Request frames and Rate Test Reply frames is described below.

#### 10.4.3.3 Interval timer

Each station should maintain a timer with a period of 128 seconds. There should be no attempt to synchronize this timer between stations. Receipt or transmission of any frames should not modify the timer. The timer interval is used when determining which nodes have been actively sending to multicast and broadcast addresses (see 10.4.4.2) and when sending reminder RRCFs in reference to multicast and broadcast addresses (see 10.4.5.1).

### 10.4.4 Sender operation

#### 10.4.4.1 Sender – Transmit Data Frame

Access the logical channel state information to determine the **sender PE** to use for transmission. Create the channel if necessary, and default the **sender PE** to PE = 33 ( Spectral Mask#2, 2-Mbaud 2 bits per symbol) if the destination node is G.9954 or PE = 1 (Spectral Mask#1, 2 bits per symbol, 2-Mbaud FDQAM) if the destination node is G.9951/2. Logical channel state information includes the node type (e.g., G.9951/2, G.9954 or unknown), the sender PE and the receiver PE for each band for which this information has been specified.

Transmissions to G.9951/2 nodes shall be sent with Spectral Mask #1 encoding.

#### 10.4.4.2 Sender – Receive Rate Change Request (RRCF OpCode 0)

For each of the RefAddrs in the RRCF (starting with RefAddr0, the SA of the RRCF frame), access the logical channel state information, if any exists, corresponding to the RefAddr and optionally RefId (further referenced by the tuple (RefAddr, [RefId]), where the square brackets indicate an optional element), and update the sender PE according to the band specification in the RRCF. If no logical channel state information exists for (RefAddr0, [RefId0]), the station should create a new logical channel state entry and initialize the sender PE according to the band specification in the RRCF. If no logical channel state information exists for additional (RefAddrs, [RefIds]), the station may either ignore those addresses or create new logical channel state entries and initialize the sender PE according to the band specification in the RRCF.

For multicast addresses and the broadcast address, senders should use a rate and Spectral Mask that is receivable by all nodes actively listening to that address. Sender stations may enforce a minimum PE which they will use to transmit to a given multicast channel, based on application-level information about QoS. It is desirable to send at the highest rate supported by the channel. Hence, if a RefAddr is a multicast address or the broadcast address, the sender should use the PE value which yields the highest raw bit rate, but which is not greater than any of the band specifications provided by the nodes actively listening to that address. Active multicast listeners shall be defined as any stations which, in either of the last two 128-second intervals:

1) either have sent any frame to the multicast address; or
2) have sent a RRCF to this station with the multicast address listed in the RefAddr list.

Active broadcast listeners shall be defined as any stations which have, in either of the last two 128-second intervals:

1) either have sent any frame to the broadcast address; or
2) sent a RRCF to this station with the broadcast address listed in the RefAddr list.

In a master-controlled network, the sender (i.e., the station at the source of the logical channel) shall update the master of the change in the negotiated PE on a Flow Channel. The master shall be

notified of the change in the flow's PE parameter by sending a Flow Modify Request with the new PE for each Flow identified in the RRCF message.

This protocol is illustrated in Figure 10-2.

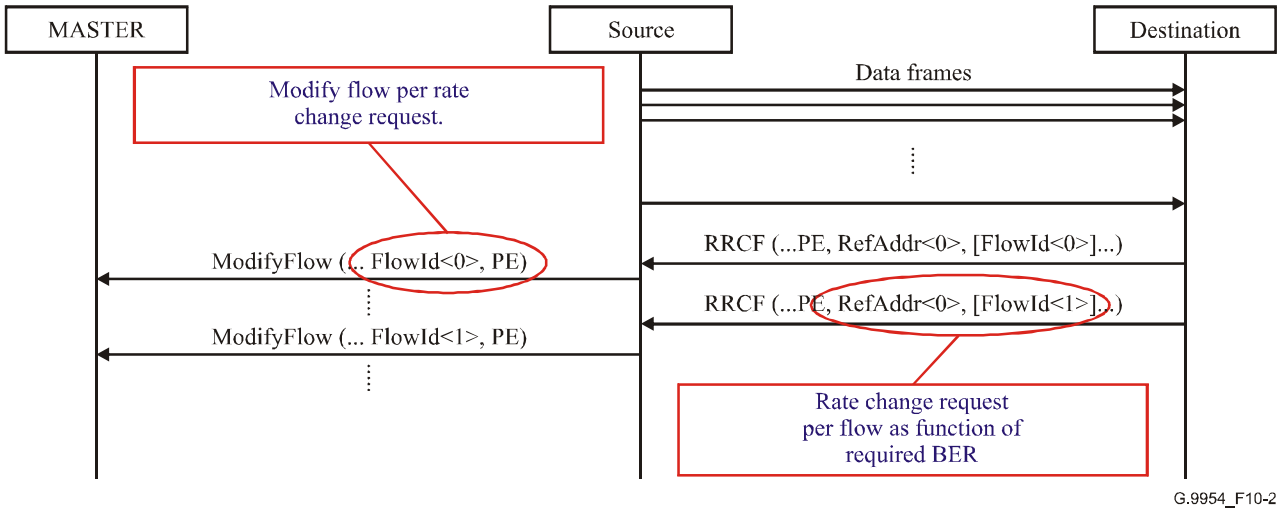


**Figure 10-2/G.9954 – Rate negotiation protocol**

### 10.4.4.3 Sender – Receive Rate Test Request frame (RRCF OpCode 1)

For each supported band encoding, generate a Rate Test Reply frame (RRCF OpCode 2) to the requestor encoded using the specified payload encoding. The contents of the RRCF shall be the current logical channel state info.

Support for Rate Test Request frames is only required in all G.9954 stations.

### 10.4.4.4 Sender – Active PNT nodes

An active PNT node is any station from which a frame has been received in either of the last two 128-second intervals.

## 10.4.5 Receiver operation

### 10.4.5.1 Receiver – Receive a Frame

The following baseline algorithm for limiting the number of RRCFs should be employed. Alternative implementations shall not generate more RRCFs than the suggested implementation. Nodes that are interested in receiving frames of a specific multicast address or of the broadcast address shall provide a mechanism to ensure that all sources of frames sent to that multicast address (or the broadcast address, as appropriate) are reminded of this node's desire to receive frames directed to that address at least once every 128 seconds (see 10.4.4.2).

For each channel, maintain a Rate Control Backoff Limit (RCBL) that ranges in value from 1 to 1024, and a Rate Control Backoff Frame Count (RCBFC) and a receiver_PE for each supported Band. (Only receiver_PE is arrayed by the number of supported Bands. RCBL and RCBFC are per channel.) RCBL is initialized to 1, and RCBFC is initialized to 0. Receiver_PE shall be initialized to 0 for Band 2. No other restrictions on receiver PE initialization are necessary. If a link integrity frame is received with PE = 1, no RRCF shall be transmitted (cf., link integrity, 10.5).

For each received frame, compute the new desired PE for the channel (new_pe) for each band. See 10.4.5.1.1 for a sample algorithm for selection of desired PE for a band. If the new desired PE is different from the previous value of the desired PE for any supported band, then reset RCBL to 1, and reset RCBFC to 0. Save the new value for desired PE (new_pe) per band, as receiver_PE. If the PE of the received frame is different from the new desired PE, then increment RCBFC by 1. If

RCBFC is now greater than or equal to RCBL, then send an RRCF to the source of the frame, with band1_PE set to receiver_PE for Band1 and Band2_PE set to receiver_PE for Band2, reset RCBFC to 0, and double RCBL up to a maximum of 1024. If a multicast or broadcast channel is active (based on receiving frames other than RRCFs within the last two 128-second intervals), and 128 seconds have passed since the receiver has sent a frame to this multicast or broadcast address, transmit an RRCF with the current receiver PE to any nodes that have sent frames to that multicast or broadcast address, with a RefAddr set to the multicast or broadcast address in question. Multiple multicast addresses may be aggregated into a single RRCF being sent to a node that has been active on multiple multicast addresses. However, only addresses for which the intended recipient of the RRCF has been active should be included.

In RRCF messages, requesting stations should attempt to specify the maximum payload encoding that they believe will have an acceptable error rate, in order to maximize the aggregate throughput of the network.

At a minimum, the 2-MBaud band shall always be specified in an RRCF.

#### 10.4.5.1.1 Sample payload encoding selection algorithm

This clause describes an example algorithm suitable for use by devices implementing a single band (Band1) on networks with additive white noise and impulse noise. Other algorithms are possible which may better optimize the selected payload encoding based on the measured channel conditions.

For each implementation, compile a table of average slicer mean squared error (ASMSE) required for each payload encoding (except PE = 8) to achieve a packet error rate (PER) of 1e-3. Define this table as DOWN_LARQ. Compile a second table with a target PER of 1e-6. Define this table as DOWN_NOLARQ. Define UP_LARQ as DOWN_LARQ with all ASMSE values decreased by 2 dB and UP_NOLARQ as DOWN_NOLARQ with all ASMSE values decreased by 2 dB.

The following steps describe how to select the new payload encoding desired for a particular channel, (new_pe), given the current payload encoding desired on that channel, (curr_pe), and a new frame is received on that channel:

1) Keep a history window of 16 G.9951/2 frames per channel. For each channel, compute the ASMSE over all frames in the history window that did not have a CRC error.

2) If in V1M2 mode, assess whether or not enough margin exists in the system to allow proper detection of compatibility frames on a per-channel basis. If, for any given channel, such margin is determined not to exist, then set new_pe = 8 for that channel. If such margin is determined to exist and curr_pe = 8, set new_pe = 1. If such margin is determined to exist and curr_pe $\neq$ 8, set new_pe = curr_pe. If this station does not support the reception of compatibility format frames, then set new_pe = 8. If new_pe = 8 or curr_pe = 8, then exit. Else:

3) If all the frames in the history window were received with a CRC error, set new_pe = 1 and exit. Else:

4) If LARQ is in use on a channel, find the greatest payload encoding in the UP_LARQ table with an ASMSE greater than or equal to the ASMSE computed in step 1. If LARQ is not in use, use the UP_NOLARQ table. Define this payload encoding as new_up_pe.

5) If LARQ is in use on a channel, find the greatest payload encoding in the DOWN_LARQ table with an ASMSE greater than or equal to the ASMSE computed in step 1. If LARQ is not in use, use the DOWN_NOLARQ table. Define this payload encoding as new_down_pe.

6) If new_up_pe > curr_pe, set new_pe = new_up_pe and exit. Else:

7) If new_down_pe < curr_pe, set new_pe = new_down_pe and exit. Else:

8) If neither 6 nor 7 is satisfied, set new_pe = curr_pe.

NOTE 1 – The offset between the up and down rate selection tables provides the algorithm with hysteresis to provide stability in selection of a payload encoding in the presence of minor variations in ASMSE. Due to this offset, conditions 6 and 7 cannot both be satisfied simultaneously.

NOTE 2 – The combination of the 16-frame history window with the selection hysteresis prevents the rate selection algorithm from generating an excessive number of rate changes while remaining responsive to significant changes in the channel conditions.

NOTE 3 – The selection algorithm for the value PE = 8 in step 2 should also include hysteresis to avoid generating an excessive number of rate changes while remaining responsive to significant changes in the channel conditions.

#### 10.4.5.2 Receiver – Send Rate Test Request frame (RRCF OpCode 1)

Periodically, but at a rate not to exceed once every 128 seconds (except as described below), a receiver may send a Rate Test Request frame to a sender to test if the channel can support a different band. The band encodings represent the encodings for which the receiver would like the sender to generate test frames. NumAddr shall be set to 0 in Rate Test Request frames.

Rate Test Request frames should be sent encoded at the current negotiated rate for the channel from the receiver to the sender.

Support for Rate Test Request frames is required in all stations.

#### 10.4.5.3 Receiver – Receive Rate Test Reply frame (RRCF OpCode 2)

Upon receipt of a Rate Test Reply frame, the receiver should use the demodulation statistics for this frame, and any previously received Rate Test Reply frames using this encoding, to make a decision as to the channel's capability to support the tested band encoding. If the decision is that the channel is not capable of supporting the tested band encoding, the receiver shall not generate another Rate Test Request frame for at least 128 seconds. If the decision is that the channel is capable of supporting the tested band encoding, the receiver may repeat the test to collect more data, at a maximum rate of one RateTest Request frame every second, with a maximum of 16 additional tests. At this point, the receiver should generate a Rate Change Request to the sender specifying the new band encoding.

Support for Rate Test Reply frames is only required in stations that implement additional bands beyond Band1. Stations that only implement Band1 may silently discard received Rate Test Reply frames.

### 10.5 Link Integrity Function

The purpose of the Link Integrity Function is to provide a means for hardware and/or software to determine whether or not this station is able to receive frames from at least one other station on the network. In the absence of other traffic, a station periodically transmits a Link Integrity Control Frame (LICF) to the Broadcast MAC address, with the interval between such transmissions governed by the method described below.

All stations shall implement the following function to ensure that, with high probability, within any 1-second interval there is:

1) either at least one LICF sent to the Broadcast MAC address from this station; or
2) at least one packet addressed to the Broadcast MAC address received from each of at least two other stations.

Additionally, all stations shall send at least one LICF every 64 seconds.

The method is described below:

- Stations SHOULD support generation of the existing LI frame even in inactive or sleep mode. While in sleep or inactive mode, PNT stations that do not want to be or cannot be awakened SHOULD not send LI frames.
- A Link Packet may be any broadcast frame received with a valid header FCS. Only LICF frames should be treated as Link Packets.
- Each station maintains a free-running timer with a period of 1 second. There should be no attempt to synchronize this timer between stations. The timer should not be modified by any link state transitions or by the reception of any frames. This timer is the source of the timeout event used in the link integrity state table in Table 10-9.
- Each station maintains a 6-bit FORCE_SEND counter that is initialized to a random value between 30 and 63. This initialization value may be selected once at node startup and used for each re-initialization of the FORCE_SEND counter, or a new random value may be selected for each re-initialization of the FORCE_SEND counter.
- Each station has a register (SA1) that can be set from the SA of a received Link Packet.
- An LICF should be sent with a priority corresponding to Link Layer priority 7 in an unmanaged network and as flow 0 in a managed network.
- The PE for an LICF shall be determined by accessing the RRCF logical channel information for the broadcast channel. An exception to this criterion is if LI frames are not sent with the currently-negotiated broadcast PE value, then they SHALL be sent with PE = 1. This allows, for example, terminals in sleep or inactive mode to maintain active status on the network. Receipt of a LI frame with PE = 1 SHALL not cause a transmission of an RRCF by any PNT terminal.
- While in sleep or inactive mode, the terminal SHOULD perform Link Integrity and wakeup processing on all receive packets. No further processing of receive packets is necessary. The relevant power-management processing shall be done on LARQ and non-LARQ data frames and the understanding is that non-WoLAN frames should be discarded.
- Each station shall send a Link Integrity Control Frame (LICF) with the format shown in Table 10-10, according to the state table in Table 10-9.

Figure 10-3 gives a pictorial view of the state transitions, with some minor loss of detail, including omission of events that do not cause state transitions (and have no associated actions), and the collapsing of multiple events into a single transition with a more complex description of the action.

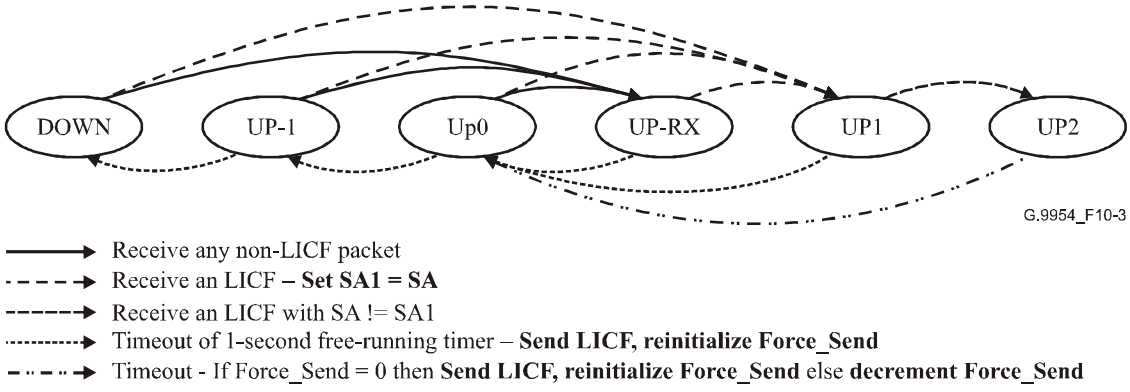


**Figure 10-3/G.9954 – Link integrity state diagram**

Table 10-9 is a complete state table, with associated actions. The timeout event is the periodic expiration of a one-second free-running timer.

Initial State: DOWN, Force_Send initialized: 30 ≤ Force_Send ≤ 63

**Table 10-9/G.9954 – Link integrity finite state machine (FSM)**

| | DOWN | UP-1 | UP0 | UP-RX | UP1 | UP2 |
|---|---|---|---|---|---|---|
| Receive any non-LICF | UP-RX | UP-RX | UP-RX | UP-RX | UP1 | UP2 |
| | (none) | (none) | (none) | (none) | (none) | (none) |
| Receive LICF with SA == SA1 | UP1 | UP1 | UP1 | UP1 | UP1 | UP2 |
| | Set SA1 ← SA | Set SA1 ← SA | Set SA1 ← SA | Set SA1 ← SA | (none) | (none) |
| Receive LICF with SA != SA1 | UP1 | UP1 | UP1 | UP1 | Native: UP2<br>Compat: UP1 | UP2 |
| | Set SA1 ← SA | Set SA1 ← SA | Set SA1 ← SA | Set SA1 ← SA | (none) | (none) |
| Timeout and Force_Send == 0 | DOWN | DOWN | UP-1 | UP0 | UP0 | UP0 |
| | Send LICF[a)], reinit Force_Send | Send LICF[a)], reinit Force_Send | Send LICF[a)], reinit Force_Send | Send LICF, reinit Force_Send | Send LICF, reinit Force_Send | Send LICF, reinit Force_Send |
| Timeout and Force_Send > 0 | DOWN | DOWN | UP-1 | UP0 | UP0 | UP0 |
| | Send LICF[a)], reinit Force_Send | Send LICF[a)], reinit Force_Send | Send LICF[a)], reinit Force_Send | Send LICF, reinit Force_Send | Send LICF, reinit Force_Send | Decrement Force_Send |

[a)] Devices which can transmit using more than one MAC source address (e.g., a bridge) should send a CSA request frame to the broadcast address instead of sending an LICF for the cases indicated in the table.

Link Integrity Status shall be indicated when in any state but DOWN. All stations should include a visible Link Status Indicator (LSI) (e.g., an LED) for indicating Link Integrity Status.

**Table 10-10/G.9954 – Link integrity short frame**

| Field | Length | Meaning |
|---|---|---|
| DA | 6 octets | Destination Address = FF:FF:FF:FF:FF:FF |
| SA | 6 octets | Source Address |
| Ethertype | 2 octets | 0x886c (PNT Link Control Frame) |
| SSType | 1 octet | = SUBTYPE_LINK (2) |
| SSLength | 1 octet | Number of additional octets in the control header, starting with the SSVersion field and ending with the second (last) octet of the Next Ethertype field. Minimum is 4 for SSVersion 0 |
| SSVersion | 1 octet | = 0 |
| LI_pad | 1 octet | Ignored on reception |
| Next Ethertype | 2 octets | = 0 |
| Pad | 40 octets | Any value octet |
| FCS | 4 octets | |
| CRC-16 | 2 octets | PNT Frame Check Sequence |

## 10.6 Capability and Status Announcement

A mechanism is defined for network-wide negotiation, capability discovery and status announcement. It is based on periodic broadcast announcements, called Capabilities and Status Announcements (CSA) sent in CSA Control Frames (CSACF). The defined status flags allow determination of a station's PNT version, optional feature support, and link-layer priority usage, as well as communication of network configuration commands.

The purpose of the protocol is to distribute to all stations the complete set of status flags in use on the network, so that stations can make operational decisions based on those flags with no further interaction.

Stations shall use the CSA Control Frame as described in Table 10-11 and the CSA Flag definitions shown in Table 10-12.

Stations shall send a CSA Control Frame once per minute or when a change in the station's current status requires the announcement of new (or deleted) flags.

A station sending a CSA Control Frame announcing a status change shall send a second copy of the most recent CSACF a short interval after the first, since it is always possible to lose a frame due to temporary changes in the channel, impulse noise, etc. The interval should be randomly selected (not simply fixed), and chosen from the range 1 to 1000 milliseconds, inclusive.

CSA Control Frames are sent with a priority corresponding to Link Layer priority 7.

CSA Control Frames are always sent to the broadcast address (0xFFFFFFFFFFFF).

The PE for a CSA control frame shall be determined by accessing the RRCF logical channel information for the broadcast channel.

A Request op-code is defined to allow a station to quickly gather complete information about all stations. Upon receiving a CSA control frame with the Request OpCode, a station shall transmit a current CSA message after a delay of a short interval, using the same mechanism (and parameters) that delays the second copy of CSA announcements, described above.

### 10.6.1 CSA Control Frame

Table 10-11 defines the format of a Capabilities and Status Announcement control frame. The first three fields beyond the Ethernet header comprise the standard header for short-format control frames.

**Table 10-11/G.9954 – Capability and Status Announcement frame**

| Field | Length | Meaning |
|---|---|---|
| DA | 6 octets | Destination Address = FF:FF:FF:FF:FF:FF |
| SA | 6 octets | Source Address, not necessarily corresponding to the MAC address to which the frame contents are applicable (See CSA_SA.) |
| Ethertype | 2 octets | 0x886c (PNT Link Control Frame) |
| SSType | 1 octet | = SUBTYPE_CSA (3) |
| SSLength | 1 octet | Number of additional octets in the control header, starting with the SSVersion field and ending with the second (last) octet of the Next Ethertype field. Minimum is 32 for SSVersion 0 |
| SSVersion | 1 octet | = 0 |
| CSA_ID_Space | 1 octet | Identifies the registration space of CSA_MFR_ID<br>0 Unspecified<br>1 JEDEC<br>2 PCI |
| CSA_MFR_ID | 2 octets | HW manufacturer ID: Identifies the manufacturer of the PHY controller chip. The purpose of this field plus the part number and revision is to identify specific implementations of the PHY specification. This is not a board or assembly-level identifier. |
| CSA_Part_No | 2 octets | HW Manufacturer Part Number: The part number of the PHY controller chip |

**Table 10-11/G.9954 – Capability and Status Announcement frame**

| Field | Length | Meaning |
|---|---|---|
| CSA_Rev | 1 octet | HW Revision |
| CSA_Opcode | 1 octet | 0 Announce<br>1 Request |
| CSA_MTU | 2 octets | Maximum size link-level PDU this receiver accepts in octets. The default value is 1526 octets. 1526 is the minimum value that shall be advertised by an PNT station. |
| CSA_SA | 6 octets | MAC address of the station to which the capabilities and status are applicable |
| CSA_device_id | 1 octet | Device ID assigned (by the master) during registration. It is reported to the PNT device with the MAC address identified in the SA field. A value of NULL_ID indicates that the device is not registered with the master.<br>NOTE – More than one station (identified by the CSA_SA) field may have the same CSA_device_id. |
| CSA_pad | 1 octet | Reserved for version 0. Shall be sent as 0, ignored on reception. Creates field alignment to 32-bit WORD boundaries. |
| CSA_CurrentTxSet | 4 octets | Configuration flags, plus all current in-use status for this station. Flag definitions are specified in Table 10-12. |
| CSA_OldestTxSet | 4 octets | A copy of the "oldest" TX flags for this station, from the period ending at least one period (minute) earlier. Flag definitions are specified in Table 10-12. |
| CSA_CurrentRxSet | 4 octets | The union of recent flags received from other stations. Flag definitions are specified in Table 10-12. |
| Next Ethertype | 2 octets | = 0 |
| Pad | | Pad to reach minFrameSize if necessary |
| FCS | 4 octets | |
| CRC-16 | 2 octets | PNT Frame Check Sequence |

### 10.6.2 Status, Configuration, Option and Priority flags

The flags in Table 10-12 shall be used for CSA_CurrentTxSet, CSA_OldestTxSet, and CSA_CurrentRxSet in Capabilities and Status Announcement control frames.

**Table 10-12/G.9954 – CSA flag set**

| Octet | Field | Length | Description |
|---|---|---|---|
| Flags0 | TxPriority7 | 1 | Station is (was) transmitting frames with LL priority 7. (always set) |
| | TxPriority6 | 1 | Station is (was) transmitting frames with LL priority 6. |
| | TxPriority5 | 1 | Station is (was) transmitting frames with LL priority 5. |
| | TxPriority4 | 1 | Station is (was) transmitting frames with LL priority 4. |
| | TxPriority3 | 1 | Station is (was) transmitting frames with LL priority 3. |
| | TxPriority2 | 1 | Station is (was) transmitting frames with LL priority 2. |
| | TxPriority1 | 1 | Station is (was) transmitting frames with LL priority 1. |
| | TxPriority0 | 1 | Station is (was) transmitting frames with LL priority 0. (always set) |
| Flags1 | Reserved | 2 | Shall be sent as 0 and ignored by stations when received. |
| | Highest Mask # Supported | 2 | Highest Mask # supported by transmitter. Support for Mask N assumes full support for all bauds in Spectral Mask N – 1.<br>0 Spectral Mask #1<br>1 Spectral Mask #2<br>2 Spectral Mask #3 |
| | Supports Frame Bursting | 1 | This station supports frame bursting. |
| | Supports Short Control Information | 1 | This station supports the reception of short control information in Frame Burst (Aggregation) format. |
| | Reserved | 1 | Reserved for legacy use |
| | Reserved | 1 | Reserved for legacy use |
| Flags2 | Frame Burst Packet Limit | 3 | 0 No Limit (actually limited by maximum link-level frame size in the highest PE)<br>1 This station supports bursts of up to 16 frames.<br>2 This station supports bursts of up to 32 frames.<br>3 This station supports bursts of up to 64 frames.<br>4 This station supports bursts of up to 128 frames.<br>5 This station supports bursts of up to 256 frames. |
| | Frame Burst Size Limit | 3 | 0 No Limit (actually limited by maximum link-level frame size in the highest PE)<br>1 This station supports bursts of up to 8K bytes.<br>2 This station supports bursts of up to 16K bytes.<br>3 This station supports bursts of up to 32K bytes.<br>4 This station supports bursts of up to 64K bytes.<br>5 This station supports bursts of up to 80K bytes.<br>For the purpose of burst size limitations, a burst consists of all the link layer frame (i.e., all the frame excluding the physical layer preamble, frame-control, pad and EOF). For further information on frame bursting and aggregation, see 10.13. |

**Table 10-12/G.9954 – CSA flag set**

| Octet | Field | Length | Description |
|---|---|---|---|
| Flags2 | Synch Mode | 1 | This station is operating in Synchronous MAC mode and is currently synchronized with the master MAC cycle.<br>0 The station is NOT operating in Synchronous MAC mode.<br>1 The station is operating in Synchronous MAC mode. |
| | Reserved | 1 | Shall be sent as 0 and ignored by stations when received. |
| Flags3 | ConfigG.9951/2 | 1 | Force use of G.9951/2 mode |
| | ConfigV1M2 | 1 | Reserved for legacy usage |
| | ConfigV1 | 1 | Reserved for legacy usage |
| | ConfigG.9954 | 1 | Force use of G.9954 mode, defers to ConfigG.9951/2 |
| | Reserved | 1 | Shall be sent as 0 and ignored by stations when received |
| | Highest Version | 3 | This station's highest supported PNT version:<br>0x000 Reserved<br>0x001 Reserved for legacy usage<br>0x010 G.9951/2<br>0x011 G.9954<br>0x0100-0x111 Reserved |

Thirty-two bit-flags shall be supported for announcing status and configuration information. The flags are divided into three basic groups: mode selection flags including PNT version information, supported options, and in-use TX link layer priority announcements. These flags shall be added to the global state as soon as announced, and removed when no longer announced by any station, either through explicit deletion or by timing them out. An in-use TX link layer priority shall be announced for a period of one to two minutes after the last frame actually sent with the priority, until the aging mechanism causes it to be deleted from CurrentTxSet.

The **default set** of status flags, used to initialize the **NewTxSet** (defined below), is defined to be the priorities 0 and 7, the station's PNT version, and any supported options.

### 10.6.3 Terms and parameters

#### 10.6.3.1 Capabilities and Status Period (CS period)

The basic time interval used to age out non-persistent status information shall be one minute. Each station has a repeating timer set to this interval. The timers in different stations are not synchronized, and synchronization should, in general, be avoided. The description below refers to the time between one expiration of this timer and the next as a "period". The "current" period refers to the time since the most recent expiration of the timer.

A CSA frame shall be sent at the end of each interval.

#### 10.6.3.2 Variables, etc.

– DeleteSet: A computed value used to detect newly removed status information.

– NewRxFlags, ReallyNewRxFlags: Computed values used to detect new status flags.

#### 10.6.3.3 Timers

– CSP_Timer: A free-running timer with a period of 60 seconds.

– RetransmitTimer: A one-shot timer, set to a random interval in the range 1 ms to 1000 ms, inclusive, after sending a CSA in which CSA_CurrentTxSet and CSA_OldestTxSet are different, or when a CSA is received with the CSA_Opcode set to 1 (Request). This timer is cancelled if a second CSA is sent as a result of the CSP_Timer expiring.

### 10.6.4 Status and Priority Set State Variables

Each station maintains five basic sets of status and priority information. In addition, three more composite sets are defined as the union of two or more of the basic sets. (See Table 10-13.)

**Table 10-13/G.9954 – Set state variables**

| | |
|---|---|
| NewTxSet | The set of flags announced during the current CS period, updated immediately when a new link layer priority is used or a new volatile status is set. When the CSP_Timer expires, CurrentTxSet is given the value of NewTxSet, and NewTxSet is reset to the default set. |
| PreviousTxSet | The set of flags that were announced during the previous CS period (the ending value of NewTxSet from the previous CS period) |
| OldestTxSet | The set of flags rolled over from PreviousTxSet at the end of the previous CS period (the value of PreviousTxSet from the previous CS period). Flags that are present in OldestTxSet and missing from PreviousTxSet were not actively used or detected (by the sender) for an entire CS period, and will be deleted. This set is sent in CSA frames as CSA_OldestTxSet. |
| NewRxSet | The union of all CSA_CurrentTxSet flags received in CSAs from other stations during the current CS period. This is rolled over into PreviousRxSet at the expiration of the CSP_Timer, then reset to the empty set (0).<br>A volatile status flag (one of the priority flags) in this set may subsequently be deleted if the only station previously announcing that flag stops using it. The deletion from that station's CurrentTxSet is noted by the difference from its OldestTxSet. The fact that it was the only sender is noted by the absence of the flag in that station's CurrentRxSet, indicating that it has received the flag from no other stations.<br>If deleted from NewRxSet, a flag shall also be deleted from PreviousRxSet. |
| PreviousRxSet | The set of announced flags received during the previous CS period (the ending value of NewRxSet from the previous CS period). A flag may be deleted from this set, as described under NewRxSet above. |
| CurrentTxSet | The set of flags that were announced during the previous CS period plus any new status and priority flags (or changed configuration/options flags) used during the current CS period, i.e., the union of PreviousTxSet and NewTxSet. This set is sent in CSA frames as CSA_CurrentTxSet. |
| CurrentRxSet | The union of NewRxSet, PreviousRxSet. This set is sent in CSA frames as CSA_CurrentRxSet. |
| CurrentInUseSet | The union of CurrentTxSet and CurrentRxSet. This set is used to determine the operational mode of the station and to modify the mapping between the LL priority of the frame and the actual PHY priority usage. |

### 10.6.5 Capabilities and Status Announcement Protocol Operation

#### 10.6.5.1 New Transmit Frame – Priority detection

The CSA Protocol does not directly process transmit frames. When the LARQ protocol is in use, CSA looks at the **LL priority** of the frame as it would normally be sent to the driver.

1) If the **LL priority** is not already in NewTxSet, add it to NewTxSet.
2) If the **LL priority** was not already in NewTxSet and it is not in PreviousTxSet, then send a new CSA control frame with the CSA_Opcode set to 0 (Announce), and start the

RetransmitTimer. If the timer was already running, then cancel and restart it. Update the current PHY priority mapping function for the driver.

#### 10.6.5.2 Receive CSA Control Frame

The receiver may want to save a copy of some or all of the most recent CSA from each other station as a simple way of tracking other station's capabilities and status.

1) Record (optionally) the status and options flags from the CSA_CurrentTxSet in a table indexed by the CSA_SA address. The options flags are used to select use of optional functions between pairs of stations that implement the same options.
2) If the CSA_Opcode in the frame is 1 (Request), then start the RetransmitTimer. If the timer is already running it should be left running, although this is not required and cancellation followed by restart is allowed.
3) If CSA_CurrentTxSet has a flag not already in NewRxSet, then add the flag to NewRxSet, and check to determine if this flag is not present in the PreviousRxSet. The corresponding boolean expressions are as follows:

   NewRxFlags = (CSA_CurrentTxSet & ~NewRxSet)

   NewRxSet |= NewRxFlags

   ReallyNewFlags = NewRxFlags & ~(PreviousRxSet | CurrentRxSet)
4) Compare CSA_OldestTxSet with CSA_CurrentTxSet. If a flag has been deleted, and if that flag is also missing from CSA_CurrentRxSet, then delete the flag from NewRxSet, and PreviousRxSet. The corresponding boolean expressions are as follows:

   DeleteSet = (CSA_OldestTxSet & ~CSA_CurrentTxSet) & ~CSA_CurrentRxSet

   NewRxSet = NewRxSet & ~DeleteSet

   PreviousRxSet = PreviousRxSet & ~DeleteSet

   CurrentRxSet = NewRxSet | PreviousRxSet
5) If either ReallyNewFlags or DeleteSet are non-zero, then update the network mode and priority mapping, as necessary.

#### 10.6.5.3 CSP_Timer timeout

When a CSP_Timer timeout occurs, a new CS period has begun. Roll over the various status sets, re-compute the composite sets, and send a CSA. Set the RetransmitTimer, if needed.

1) OldInUseSet = CurrentInUseSet.
2) Move NewRxSet to PreviousRxSet.
3) Set NewRxSet to 0 (empty set).
4) Move PreviousTxSet to OldestTxSet.
5) Move NewTxSet to PreviousTxSet.
6) Set NewTxSet to the default set, consisting of this station's highest supported version, current configuration flags if any (normally none), currently supported options, and the default priority set {0,7}.
7) Update CurrentTxSet, CurrentRxSet, and CurrentInUseSet (at least logically, an implementation need not keep separate copies of these values).

   CurrentRxSet = NewRxSet | PreviousRxSet.

   CurrentTxSet = NewTxSet | PreviousTxSet.

   CurrentInUseSet = CurrentRxSet | CurrentTxSet.
8) Send a CSA frame with the CSA_Opcode set to 0 (Announce), including the updated flags.

9) If CSA_CurrentTxSet and CSA_OldestTxSet in the CSA frame just sent were different, start the RetransmitTimer. If the timer was previously running, then cancel it and restart it.

10) If one or more status flags have been deleted, then recompute the network operating mode and/or priority mapping function due to changed status flags. The mode/mapping recomputation should be performed if CurrentInUseSet is not equal to OldInUseSet.

#### 10.6.5.4 Retransmit timeout

If the RetransmitTimer expires, send a current CSA frame for this station with the CSA_Opcode set to 0 (Announce). The timer shall not be restarted.

### 10.6.6 Network mode selection based on CurrentInUseSet

The mode selection flags of the CSA protocol (configG.9951/2, configG.9954) are intended to be employed by higher-layer entities which make mode switching decisions, such as user interfaces or test utility control functions.

### 10.6.7 Priorities

There is a cost of slightly lower maximum attainable bandwidth associated with lower PHY priorities in the PNT MAC protocol if a default mapping scheme of link layer to PHY layer priorities is employed. This cost becomes especially burdensome when only lower-priority traffic is being carried on the network. Therefore, the CSA protocol includes procedures for remapping lower LL priorities to higher PHY layer priorities when no station on the network is sending traffic marked for those higher priorities.

The choice of Physical Layer (PHY) priority for a given frame is based on its assigned Link Layer (LL) priority. The default mapping from LL priority to PHY priority is specified in 10.6.7.3. The LL priority of a frame at the sender must be conveyed to the receiving station in order to allow proper recovery of link layer protocol at the receiver. This requires either a fixed, one-to-one, mapping of LL to PHY priorities, or some mechanism for carrying the LL priority within each frame. The LARQ protocol, defined in 10.7, carries the assigned LL priority from a sending station to a receiving station, providing the required mechanism, and thereby creating the opportunity to apply non-default LL to PHY priority mappings, which in turn, allows for higher maximum attainable bandwidth. A station may optionally use an 802.1q header to convey the LL priority. However, since support for 802.1q headers is optional, a station employing this method should attempt to determine that all receivers of the frame support the use of 802.1q headers. Stations that do not support 802.1q headers are unlikely to properly receive frames that include an 802.1q header.

#### 10.6.7.1 Transmit frames – Choice of physical priority

When the assignment of a Physical layer priority to the frame occurs, any changes to the PHY priority remapping function due to the use of a new priority should already have been made. The driver should use the remapped PHY priority to transmit the frame (including placing this value in the Frame Control Header) unless the frame has no LARQ header, in which case the default LL-to-PHY mapping shall be used.

#### 10.6.7.2 Received frame priorities

The LL priority of received frames indicated up the protocol stack by the driver (before any reassignment due to a LARQ or 802.1q header) shall be determined using the default PHY-to-LL priority map. The mechanism that guarantees correct LL priority for received frames is the restoration of LL priority from the LARQ (or optionally, 802.1q) header or from the Flow Specification. LARQ header processing shall be performed after the default LL priority has been assigned in the receive path. If a received frame can be mapped to a Flow Channel, the priority information in the associated Flow Specification shall be used to recover the LL priority.

### 10.6.7.3 Default Link Layer to Physical Layer Map

The IEEE 802.1p specification places the default (unassigned/best-effort) priority above both priorities 1 and 2, when an 8-level priority system is in use. Therefore, Link Layer priority 0 shall be mapped above both LL 1 and LL 2 for default Physical Layer priority assignment. IEEE 802.1p designates priority level 7 for Network Control and priority level 6 for traffic requiring latency of <10 ms (typically characterized as voice-like traffic). However, on PNT networks, PHY priority level 7 shall be reserved for traffic requiring latency of <10 ms, and Network Control traffic shall be redirected to PNT PHY priority level 6. Link layer priority 5 shall be reserved for traffic requiring latency of <100 ms. So the default mapping for LL to PHY priorities includes the swapping of priorities 6 and 7.

For transmitted frames, the set of LL priorities [0,1,2,3,4,5,6,7] shall be mapped, by default, in order to the following set of PHY priorities [2,0,1,3,4,5,7,6].

For received frames, PHY priorities [0,1,2,3,4,5,6,7] shall be mapped, by default, to LL priorities [1,2,0,3,4,5,7,6].

### 10.6.8 Priority mapping and LARQ

The PHY priority remapping shall be performed below LARQ in the protocol stack, and shall not be applied to the priority field in the LARQ (or optionally, 802.1q) header. PHY priority remapping shall not be performed on data frames (those that are not link control frames) unless a LARQ (or optionally, 802.1q) header has been added with the original LL priority. PHY priority remapping shall be performed on Link Control Frames.

### 10.6.9 Priority remapping based on CurrentInUseSet

Without priority mapping, a station would pass the original **LL priority** into the driver, where that value would be used to select the associated PHY priority from the default map. With priority remapping, the **default-assigned PHY priorities** are increased to make use of higher PHY priorities that would otherwise be unused. The remapping function is simple. For each PHY priority **P** that corresponds to an in-use LL priority, the new priority **P'** to use shall be that priority increased by the number of higher unused priorities. For example, if [1,3,4,7] are in use, then priority 4 will be increased by 2 to 6, since there are two higher unused priorities (5,6). Figure 10-5 contains a few more examples that should make this clear (including the default LL-to-PHY translation). The columns in Figures 10-4 and 10-5 represent **LL priorities** before mapping. The left-hand section shows some sets of in-use priorities, with the right-hand section showing the new PHY priority that the driver should use in each case.

| | | | | | | | | | TX LL priority | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| CurrentInUse priorities (any) | | | | | | | | | Default TX PHY priorities | | | | | | | |
| a | n | y | t | x | s | e | t | | 2 | 0 | 1 | 3 | 4 | 5 | 7 | 6 |

**Figure 10-4/G.9954 – Default LL to PHY TX priority mapping**

| | | | | | | | | | TX LL priority | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| CurrentInUse priorities (LL) | | | | | | | | | Remapped TX PHY priorities | | | | | | | |
| 0 | | | | | | | 7 | | 6 | 5 | 5 | 6 | 6 | 6 | 7 | 7 |
| 0 | | | | | | 6 | 7 | | 5 | 4 | 4 | 5 | 5 | 5 | 7 | 6 |
| 0 | 1 | | | 4 | | | 7 | | 5 | 4 | 4 | 5 | 6 | 6 | 7 | 7 |
| 0 | | | 3 | | 5 | 6 | 7 | | 3 | 2 | 2 | 4 | 4 | 5 | 7 | 6 |

**Figure 10-5/G.9954 – Direct LL to PHY TX priority remapping**

The shaded entries in Figure 10-5 show mappings that no sender should be using. However, if there is any possibility of an implementation sending with an out-of-date mapping, or sending a priority that has not been included in the mapping, then it should always use the priority of the next lower valid mapping.

Here is one example in detail. If the CurrentInUse are [0,1,4,7], then the corresponding set of in-use PHY priorities is [2,0,4,6]. Then increase each by the number of missing higher priorities: 2→5, 0→4, 4→6 and 6→7. Just to be safe, the any unused PHY priorities are also remapped to the new value of the next lower in-use priority, giving: 1→4, 3→5, 5→6, 7→7.

So the in-use LL priorities [0,1,4,7] result in transmitting PHY priorities [5,4,6,7]. A complete map for all the LL priorities adds the remaining remapped values for the default priorities corresponding to the unused LL priorities: LL[0,1,2,3,4,5,6,7] gives PHY[5,4,4,5,6,6,7,7].

## 10.7 LARQ: Limited Automatic Repeat Request Protocol

Limited Automatic Repeat reQuest (LARQ) is a protocol that reduces the effective error rate when frame errors occur. Its primary distinction from similar, sequence number-based protocols is that it does not guarantee reliable delivery of every frame, but instead conceals errors in the physical layer through fast retransmission of frames. The goal is to significantly enhance the usability of networks that may, at least occasionally, have frame error rates (FER) of 1 in $10^{-2}$ or worse. Protocols such as TCP are known to perform poorly when FER gets high enough, and other applications, such as multimedia over streaming transport layers, are also susceptible to poor performance due to high FER conditions.

The protocol provides a negative acknowledgment (NACK) mechanism for receivers to request the retransmission of frames that were missed or received with errors. There is no positive acknowledgment mechanism. There is no explicit connection setup or teardown mechanism. A reminder mechanism gives receivers a second chance to detect missing frames when relatively long gaps (in time) occur between frames.

LARQ functions is an adaptation layer between the Ethernet link layer (layer 2) and the IP network layer (layer 3). It is commonly implemented in the device driver.

Stations implement LARQ per "LARQ channel", where a LARQ channel is identified by either the tuple {source address, destination address, priority}, referred to as a LARQ-Priority channel or by the tuple {source address, destination address, flow id}, referred to as a LARQ-Flow channel.

LARQ-Priority channel is defined (and set up) in an implementation-dependent way. A LARQ-Flow channel is defined when the ACK-Policy for the associated Flow (in the Flow Specification) is set to "LARQ" and set up in conjunction with the setup of the flow.

Stations may enable or disable LARQ processing on a channel dynamically, based on information about network frame error rates. However, LARQ should be left enabled at all times, since the per-packet processing overhead is quite low, and the complexity associated with enabling and disabling the protocol (including determination of appropriate parameters) probably outweighs any likely performance gains.

Stations should implement LARQ, and if they do so, they shall use the specified control frame formats and should use the procedures defined below.

For a Simple Channel (i.e., a logical channel defined by SADA without an associated flow specification) station not adding LARQ (or optionally, 802.1q) headers shall not remap PHY priorities, and shall treat all received traffic as "best effort", that is, all traffic shall be assigned to Link Layer Priority 0. For a Flow Channel (i.e., logical channel defined by SADAflow ID) PHY priority remapping and LL priority recovery is performed using the priority information in the flow specification.

Stations may choose to add LARQ headers on transmitted frames with the LARQ_NoRtx flag set to 1. This flag indicates that the station does not retransmit frames for this channel, but adding the LARQ header allows the station to use PHY priority remapping since the LL priority of successfully received frames will be restored from the LARQ header.

All stations SHALL be capable of removing LARQ headers from received frames (de-encapsulating the original payloads). Furthermore, if the implementation supports multiple LL priorities in its receive protocol processing, then it shall restore the LL priority from the LARQ header, if one is present. If a station does not implement LARQ, then it shall drop LARQ control frames and it shall discard frames marked as retransmissions in the LARQ header.

### 10.7.1 Frame formats – Encapsulating headers

The text below uses the terms "insert" and "remove" when discussing LARQ headers. The formal definition of the LARQ frame format provides a Next Ethertype field that contains the original frame's Ethertype value. In practice, it will generally be the case that LARQ frames will be created by inserting the 8 octets starting with the Ethertype 0x886c into the original frame between the Ethernet header's source address and the original frame's Ethertype. The original frame's Ethertype becomes relabeled as the Next Ethertype field of the final frame.

The LARQ header carries LLC priority across the network. The use of 802.1q headers is not required for this function, and PNT drivers are not required to support the use of 802.1q headers for conveying priority.

**Table 10-14/G.9954 – LARQ reminder control frame**

| Field | Length | Meaning |
|---|---|---|
| DA | 6 octets | Destination Address |
| SA | 6 octets | Source Address |
| Ethertype | 2 octets | 0x886c (PNT Link Control Frame) |
| SSType | 1 octet | = SUBTYPE_LARQ (4) |
| SSLength | 1 octet | Number of additional octets in the control header, starting with the SSVersion field and ending with the second (last) octet of the Next Ethertype field. SSLength is 6 for SSVersion 0. |
| SSVersion | 1 octet | = 0 |
| LARQ_hdr data | 3 octets | LARQ Control Header data with LARQ_ctl bit = 1, LARQ_NACK = 0 |
| Next Ethertype | 2 octets | = 0 |
| Pad | 38 octets | |
| FCS | 4 octets | Frame Check Sequence |
| CRC-16 | 2 octets | PNT Frame Check Sequence |

**Table 10-15/G.9954 – LARQ NACK control frame**

| Field | Length | Meaning |
|---|---|---|
| DA | 6 octets | Destination Address |
| SA | 6 octets | Source Address |
| Ethertype | 2 octets | 0x886c (PNT Link Control Frame) |
| SSType | 1 octet | = SUBTYPE_LARQ (4) |
| SSLength | 1 octet | Number of additional octets in the control header, starting with the SSVersion field and ending with the second (last) octet of the Next Ethertype field. SSLength is 12 for Nack frames with SSVersion 0. |
| SSVersion | 1 octet | = 0 |
| LARQ_hdr data | 3 octets | LARQ Control Header data with LARQ_ctl bit = 1, LARQ_NACK = 1..7 |
| NACK_DA | 6 octets | Original Destination Address |
| Next Ethertype | 2 octets | = 0 |
| Pad | 32 octets | |
| FCS | 4 octets | Frame Check Sequence |
| CRC-16 | 2 octets | PNT Frame Check Sequence |

**Table 10-16/G.9954 – LARQ encapsulation frame**

| Field | Length | Meaning |
|---|---|---|
| DA | 6 octets | Destination Address (from original Ethernet PDU) |
| SA | 6 octets | Source Address (from original Ethernet PDU) |
| Ethertype | 2 octets | 0x886c (PNT Link Control Frame) |
| SSType | 1 octet | = SUBTYPE_LARQ (4) |
| SSLength | 1 octet | Number of additional octets in the control header, starting with the SSVersion field and ending with the second (last) octet of the Next Ethertype field. SSLength is 6 for SSVersion 0 = 6 |
| SSVersion | 1 octet | = 0 |
| LARQ_hdr data | 3 octets | LARQ Encapsulation header data (with LARQ_ctl bit = 0) |
| Next Ethertype | 2 octets | From original Ethernet PDU |
| Payload | Min. 46 octets | From original Ethernet PDU payload |
| FCS | 4 octets | Frame Check Sequence |
| CRC-16 | 2 octets | PNT Frame Check Sequence |

**Table 10-17/G.9954 – LARQ encapsulation header data**

| Octet | Field | Length | Meaning |
|---|---|---|---|
| Flags0 | LARQ_Mult | 1 bit | Multiple Retransmission Flag. 0 in the original transmission of a data frame. For retransmitted frames (LARQ_Rtx = 1), set to the value of LARQ_Mult in the NACK frame that caused the retransmission. This flag can be used by receivers to measure the round-trip times associated with the miss/Nack/receive-rtx process. |
| | LARQ_Rtx | 1 bit | 0 for first transmission of a frame, 1 if frame is retransmitted. Stations not implementing LARQ shall drop any data frame if this bit is 1. |
| | LARQ_NewSeq | 1 bit | 1 if the sequence number space for the channel has been reset, and older sequence numbers should not be nacked, 0 otherwise. |
| | LARQ_NoRtx | 1 bit | 0 if implementation supports retransmission, 1 if only priority is meaningful. May be used on a per-channel basis. |
| | LARQ_Ctl | 1 bit | "0" when in Encapsulation Format |
| | Priority | 3 bits | Link Layer Priority/Flow ID of this frame |
| Flags1_Seq0 | Reserved | 2 bits | Reserved for future use |
| | LARQ_seq_high | 4 bits | High 4 bits of Sequence number |
| Seq1 | LARQ_seq_low | 8 bits | Low 8 bits of Sequence number |

The exact application of the LARQ_Rtx, LARQ_NewSeq and LARQ_NoRtx bits requires further explanation as found in Table 10-18.

**Table 10-18/G.9954 – LARQ_Rtx, LARQ_NewSeq and LARQ_NoRtx bits interpretation**

| LARQ_Rtx | LARQ_NewSeq | LARQ_NoRtx | Interpretation |
|---|---|---|---|
| 0 | 0 | 0 | Normal transmission on an active channel<br>This combination is used for the first transmission of a frame on an active LARQ channel.<br>The receiver of this frame should send NACKs for earlier sequence numbers that are determined to be missing when this frame is received, or for this frame, if this frame has a CRC error but the LARQ header appears to be in sequence for the channel. |
| 0 | 0 | 1 | Used for the first transmission of a frame which will not be retransmitted in response to a NACK<br>The sender should use this combination when it does not save the frame for retransmission in response to receiving a NACK.<br>If a receiver is keeping state, then it should send this frame up when it has either received frames for all previous sequence numbers, or given up attempts to receive frames for all previous sequence numbers. |

**Table 10-18/G.9954 – LARQ_Rtx, LARQ_NewSeq and LARQ_NoRtx bits interpretation**

| LARQ_Rtx | LARQ_NewSeq | LARQ_NoRtx | Interpretation |
|---|---|---|---|
| 0 | 1 | 0 | Used for the first transmission of a frame with a new sequence number space<br>The sender uses this combination when there are no saved frames for the channel, excepting this frame.<br>The receiver should send all frames for this channel up to the next layer, since there is no longer the possibility of receiving any frames with previous sequence numbers. The receiver of this frame should send a NACK for this frame, if this frame has a CRC error but the LARQ header appears to be in sequence for the channel. |
| 0 | 1 | 1 | Used for the first transmission of a frame with a new sequence number space which will not be retransmitted in response to a NACK<br>The sender uses this combination when there are no saved frames for the channel.<br>The receiver should send all frames for this channel up to the next layer, since there is no longer the possibility of receiving any frames with previous sequence numbers. |
| 1 | 0 | 0 | Retransmission of a frame for this channel<br>Sender uses this combination to send a frame which has been transmitted before, and for which a NACK will cause an additional retransmission.<br>The receiver must accept this frame if it is not a duplicate. If the receiver is not maintaining state for the channel, then this frame must be discarded because it would be impossible to determine the duplicate status for the frame. The receiver of this frame should send a NACK for this frame, if this frame has a CRC error but the LARQ header appears to be in sequence for the channel. |
| 1 | 0 | 1 | Retransmission of a frame for this channel<br>Sender uses this combination to send a frame which has been transmitted before, but has not been saved for retransmission in response to receiving a NACK.<br>The receiver must accept this frame if it is not a duplicate. If the receiver is not maintaining state for the channel, then this frame must be discarded because it would be impossible to determine the duplicate status for the frame. |

**Table 10-18/G.9954 – LARQ_Rtx, LARQ_NewSeq and LARQ_NoRtx bits interpretation**

| LARQ_Rtx | LARQ_NewSeq | LARQ_NoRtx | Interpretation |
|---|---|---|---|
| 1 | 1 | 0 | Retransmission of a frame for this channel<br>The sender uses this combination when there are no older saved frames for the channel, excepting this frame.<br>The receiver must accept this frame if it is not a duplicate. If the receiver is not maintaining state for the channel, then this frame must be discarded because it would be impossible to determine the duplicate status for the frame. The receiver should send this frame and all older frames for this channel up to the next layer, since there is no longer the possibility of receiving any frames with previous sequence numbers. The receiver of this frame should send a NACK for this frame, if this frame has a CRC error but the LARQ header appears to be in sequence for the channel. |
| 1 | 1 | 1 | Retransmission of a frame for this channel<br>The sender uses this combination when there are no older saved frames for the channel.<br>The receiver must accept this frame if it is not a duplicate. If the receiver is not maintaining state for the channel, then this frame must be discarded because it would be impossible to determine the duplicate status for the frame. The receiver should send this frame and all older frames for this channel up to the next layer, since there is no longer the possibility of receiving any frames with previous sequence numbers. |

**Table 10-19/G.9954 – LARQ control header data**

| Octet | Field | Length | Meaning |
|---|---|---|---|
| Flags0 | LARQ_Mult | 1 bit | Multiple Retransmission Flag. 0 in the first Nack sent for a given sequence number, 1 in all retransmitted Nacks. |
| | LARQ_NACK | 3 bits | NACK Count<br>If 0 in a LARQ Control Frame, then this is a Reminder. |
| | LARQ_Ctl | 1 bit | Set to 1 for LARQ Control Header data format |
| | Priority/FlowID | 3 bits | Link Layer Priority/Flow ID of this frame |

**Table 10-19/G.9954 – LARQ control header data**

| Octet | Field | Length | Meaning |
|---|---|---|---|
| Flags1_Seq0 | FlowID | 1 bit | High order bit of FlowID if FSelector = 1 |
| | FSelector | 1 bit | Select interpretation of Priority/Flow ID field.<br>0 Priority interpretation<br>1 Flow ID interpretation |
| | Reserved | 2 bits | Reserved for future use |
| | LARQ_seq_high | 4 bits | High 4 bits of Sequence number |
| Seq1 | LARQ_seq_low | 8 bits | Low 8 bits of Sequence number |

### 10.7.2 Terms and definitions

**10.7.2.1 control frame**: A frame generated by a LARQ protocol module that contains only a LARQ protocol header as its payload.

**10.7.2.2 current sequence number**: The most recently received new sequence number for a channel.

**10.7.2.3 data frame**: Any standard Ethernet frame from higher (than LARQ) protocol layers. A LARQ-enabled station encapsulates the original payload of an Ethernet frame by inserting a LARQ header (short form control header with LARQ_hdr data) between the source address and the remainder of the frame before the frame is passed down to the driver for transmission on the network.

**10.7.2.4 forget timer**: An implementation-dependent mechanism to allow a receiver to reset the sequence number space of a channel when a received sequence number is not the next expected (Current Sequence Number + 1). 1 s is a suggested default value.

**10.7.2.5 hold timer, lost timer**: An implementation-dependent timing mechanism that limits the time a receiver will hold onto a received frame while waiting for a missing frame to be retransmitted. Conceptually, there is one such timer per missing sequence number. The timer interval is Maximum Hold Interval.

**10.7.2.6 logical channel, channel**: A flow of frames from a sender to one or more receivers on a single network segment consisting of all the frames with a single combination of destination address, source address, and link layer priority or flow ID.

**10.7.2.7 NACK, Nack, nack**: An indication from a receiver to a sender requesting retransmission of one or more frames. Also, the action of providing such an indication. E.g., "to nack a sequence number" means to send a NACK indication.

**10.7.2.8 NACK timer**: An implementation-dependent timing mechanism used by a receiver to retransmit NACKs for missing sequence numbers. Conceptually, there is one such timer per missing sequence number per logical channel. The timer is reset each time a NACK is sent for a sequence number. The timer interval is NACK Retransmission Interval.

**10.7.2.9 new**: A new sequence number is one whose difference from the current sequence number for the channel, modulo the size of the sequence number space and considered as a signed integer, is greater than 0. In particular, the numbers (current + 1) through (current + 2047).

**10.7.2.10 old**: An old sequence number is one whose difference from the current sequence number for the channel, modulo the size of the sequence number space and considered as a signed integer, is less than or equal to 0. In particular, the numbers (current – 2048) through (current) are old. However, most of the old sequence numbers are also out of sequence.

**10.7.2.11 out of sequence**: Any sequence number that falls outside a reasonable range, old or new, of the current sequence number for a logical channel is considered out of sequence. Plus or minus twice the value of MaximumSaveLimit (defined below) should be used as the "reasonable range" when checking for out of sequence.

**10.7.2.12 receiver**: A station that receives frames sent on a particular channel. If the destination address is a unicast address there is at most one receiver. If the destination address is a group address (including broadcast), then there may be many receivers.

**10.7.2.13 reminder**: A control frame sent by the channel sender with the most recently used sequence number for a channel which has been inactive for Reminder Interval after its most recent data frame.

**10.7.2.14 reminder timer**: An implementation-dependent timing mechanism used by a sender to generate a reminder frame after a period of inactivity for a channel. The timer is reset each time a new data frame is transmitted. Conceptually, there is one such timer per channel. The timer interval is Reminder Interval.

**10.7.2.15 save timer**: An implementation-dependent timing mechanism that limits the time a sender will save a frame waiting for retransmission requests. The timer interval is Maximum Save Interval.

**10.7.2.16 sender**: The sending station for a channel, usually the station owning the source MAC address.

**10.7.2.17 sequence numbers**: Sequence numbers are maintained separately for each logical channel by the sender.

### 10.7.3 Channels

LARQ is defined for operation on simplex logical channels. A separate logical channel is defined for each combination of Ethernet destination address, Ethernet source address and link layer priority or Ethernet destination address, Ethernet source address and Flow ID. There is no explicit channel setup procedure. A new channel is implicitly defined when a station chooses to send LARQ encapsulated frames for a new combination of DA, SA and link layer priority or Flow ID. For a flow channel, an associated LARQ channel may be implicitly set up when the flow is set up if the ACK policy defined for the flow is LARQ.

The station that sends such LARQ encapsulated frames (usually the owner of the SA, except in the case of a bridge masquerading as SA) is the **sender** for the channel. Each channel has a single **sender**. Any station that receives the frames and processes the LARQ headers is a **receiver**. There may be any number of **receivers**. **Receivers** operate independently.

### 10.7.4 Sender operation

#### 10.7.4.1 Variables and parameters

- Send Sequence Number: The sequence number of the most recently transmitted data frame.
- Reminder Timer Interval: A fixed interval. The default is 50 ms. Lower values will increase the overhead of reminders on network load, while higher values increase the latency for end-of-sequence frames requiring retransmission. Implementations should not use values outside of the range 25-75 ms, based on 150-ms maximum save and hold times.
- Minimum Retransmission Interval: An interval used to prevent too-frequent retransmissions of a single frame. Most important for multicast channels. The default is 10 ms.

- Maximum Save Limit: The maximum number of frames that will be saved for a single logical channel. This is implementation dependent, and varies with the maximum frame rate the sender is expected to support. Values of 100 or more can be useful for high-speed applications such as video.
- Maximum Save Interval: The maximum time that the sender will normally save a frame for possible retransmission. The default is 150 ms.

#### 10.7.4.2 Sender – New channel

Select implementation-dependent parameters, if necessary.

Select an initial value for **Send Sequence Number**.

#### 10.7.4.3 Sender – Transmit new data frame

The link layer priority for the frame is determined in an implementation-dependent manner, for instance, by examining the 802.1p priority passed along with packets in newer NDIS implementations.

Access the logical channel state information for the DA, SA and link layer priority/Flow ID of the frame.

Increment Send Sequence Number, modulo 4096 (the size of the sequence number space).

Build the LARQ header with the new value of Send Sequence Number, and the Multiple Retransmission flag set to 0. The Priority field in the LARQ header shall be set to the Link Layer priority value specified for the frame. If no priority is specified, then the priority shall be set to 0. The method of specifying priority and the choice of value are implementation dependent and outside the scope of this Recommendation for LARQ-Priority Channels. For LARQ-Flow Channels, the LL priority shall be set using the priority specified in the flow specification.

Insert a LARQ header (short form control frame format with LARQ_hdr data) between the SA and the Ethertype/Length field of the original frame. The new frame is eight bytes longer than the original.

Save a copy of the frame.

Send the frame.

Restart the reminder timer for the channel.

Start a save timer for the sequence number. When no other resource limitations apply, a sending station should normally save a frame for Maximum Save Interval, which corresponds to Maximum Hold Interval used by LARQ receivers.

#### 10.7.4.4 Sender – Process a NACK Control Frame

The Priority/Flow ID and Original Destination Address (NACK_DA) are read from the LARQ NACK header.

Access the logical channel state information for the Sender channel, where the channel DA is the NACK_DA and the channel SA is the Ethernet DA from the Nack control frame.

The NACK Count in the LARQ header indicates the number of sequence numbers requested for retransmission. The first indicated sequence number is the value Sequence Number in the NACK header, followed by the next (NACK Count – 1) sequence numbers. For each indicated sequence number starting with the first:

– If a copy of the original frame is no longer available, go to the next sequence number.

– If the most recent retransmission of the frame is within Minimum Retransmission Interval of the current time, go to the next sequence number.
– Prepare a copy of the original frame with its original LARQ header for retransmission.
– Copy the value of the Multiple Retransmission Flag from the NACK header into the LARQ header of the frame to be retransmitted.
– Set the LARQ_Rtx flag to 1.
– Send the retransmitted frame.

Do not send a retransmission if a received Nack control frame has an error.

#### 10.7.4.5 Sender – Reminder Timer expiration

If the reminder timer expires, create a Reminder control frame, with the Sequence Number set to the current value of Send Sequence Number for the channel. The priority for the Reminder control frame shall be the same as the priority for the channel.

Send the frame.

Do not restart the reminder timer for the channel.

#### 10.7.4.6 Sender – Save Timer expiration

The save timer is implementation dependent. Its purpose is to set an upper bound on how long frames will be saved by a sender for possible retransmission. If set too long, host resources may be wasted saving frames that will never be retransmitted.

This timer is conceptually implemented per sequence number. Release any resources associated with the saved frame.

#### 10.7.4.7 Sender – Resource management

A LARQ implementation requires careful attention to resource management. The resources include the buffers used for saving copies of data for retransmission, the buffers and other resources used to manage the re-ordering of frames to incorporate retransmissions, and the various timers used to govern proper behaviour and efficient protocol operation. Resource management is implementation dependent. However, the following guidelines should be followed.

Saved copies of frames should be kept for Maximum Save Interval (default is 150 ms), other considerations notwithstanding.

Maximum Save Limit, the maximum number of saved frames for any channel, should be a function of the maximum rate that new frames may be generated. Very slow devices might usefully save only a couple of frames for retransmission. A high-speed device serving video streams might save 100 or more frames for a single channel.

Senders that save relatively few frames are more likely to receive NACK control frames for sequence numbers that can no longer be retransmitted. Such behaviour is inefficient, but causes no other problems.

### 10.7.5 Receiver operation

#### 10.7.5.1 Channel variables and parameters

The description below of correct protocol operation uses the following variables. The actual implementation may vary so long as the behaviour remains unchanged.

- Current Sequence Number: The most recent sequence number received in a LARQ header for the channel, whether in a data frame or a reminder control frame.

- Oldest missing sequence number: The oldest sequence number for a frame not yet received which has not been declared lost.
- Maximum Hold Interval: The longest interval that a frame will be held awaiting an earlier missing frame. The default is to use the same value as Maximum Save Interval, which has a default of 150 ms.
- Maximum Receive Limit: The maximum number of frames that a receiver will buffer while awaiting an earlier missing frame. The default should normally be the same as the Maximum Save Limit.
- NACK Retransmission Interval: The interval after which a receiver will retransmit a Nack control frame for a missing sequence number, with the expectation that earlier Nack control frames or data frame retransmissions were lost. The default for fixed implementations is 20 ms.

#### 10.7.5.2 Receiver – New channel

When a data frame with a LARQ header or a LARQ Reminder control frame is received, the receiver shall determine the identity of the LARQ channel (i.e., either {DA, SA, priority} or {DA, SA, flow id}) using information in the LARQ frame (i.e., Frame-Control and LARQ Encapsulation Header) and determine whether it is a new channel. If the LARQ channel is new, the receiver shall initialize state information for a new channel. For a Flow Channel, the associated LARQ Channel may be set up during Flow Setup if the setup flow has an ACK Policy = LARQ.

The primary piece of state information is the Current Sequence Number for the channel. Current Sequence Number shall be initialized to the sequence number immediately preceding that found in the LARQ header of the received frame. This assignment shall take place prior to processing the received frame and will result in the frame either appearing to be the next expected data frame, or the reminder for the next expected data frame.

#### 10.7.5.3 Receiver – LARQ data or reminder frame

Look up the channel state information based on the Ethernet DA and SA in the received frame plus the Link Layer priority/Flow ID from the LARQ header. (Set up a new channel if necessary.)

If the received sequence number of the received frame is out of sequence, the channel state may be reset. If the sequence number (before resetting) is old, and the Forget timer has expired, then the sequence space may be reset to the value of the received frame's sequence number.

If the received sequence number is newer than the Current Sequence Number (after any reset of the sequence number space) then perform new sequence number processing steps below; otherwise, perform the old sequence number processing steps.

#### 10.7.5.4 Receiver – LARQ frames with CRC or other errors

For best performance, implementations should allow the LARQ protocol module to process errored frames, such as those with payload CRC errors. This will, in many cases, allow Nack indications to be sent more quickly since the receiver will not have to wait for the next frame to detect the loss. At the same time, it provides a second opportunity for detecting lost frames at the end of a sequence, when a later Reminder would be the only protection.

If errored frames are used, they shall be used only to detect a very small set of missing sequence numbers for an existing channel (preferably one missed frame). In particular, if the errored frame appears to have a valid LARQ header, and the frame's source MAC address, destination MAC address, and LARQ header priority/Flow ID match an existing logical channel, and if the sequence

number is (Current Sequence Number + 1), then treat this frame as a Reminder control frame for the purposes of processing. Reminder control frames are always dropped after processing.

In all other cases, drop the errored frame with no further processing. Do not set up a new channel if the frame has an error. Do not send a retransmission if a Nack control frame has an error. Do not reset a channel (for sequence numbering purposes) for an errored frame.

#### 10.7.5.5 Receiver – New sequence number

If the frame has an error indicated by a lower layer driver, such as a CRC error, and the sequence number of the frame is anything other than (Current Sequence Number + 1), then drop the frame with no further processing. Otherwise, process the frame as a Reminder control frame.

If the difference between the new sequence number of the received frame and the oldest missing sequence number is greater than (Maximum Receive Limit – 1), then repeat the following steps until the acceptable limit is reached.

Cancel the Nack retransmission timer and the lost frame timer for the oldest missing sequence number.

If there is a saved frame for the next sequence number, then deliver in-sequence frames to the next layer above until the next sequence number with a missing frame is reached (which may be the next expected sequence number for the channel (Current Sequence Number + 1)). The value from the Priority/Flow ID field from the LARQ header for each frame is delivered to the next layer along with each associated frame. The method of specifying priority/Flow ID to the next layer is implementation dependent and outside the scope of this Recommendation.

If the sequence number is the next expected sequence number (Current Sequence Number + 1) and the frame is a good data frame and there are no older missing sequence numbers, then send the frame up to the next layer.

If the sequence number is newer than (Current Sequence Number + 1), or is a reminder for (Current Sequence Number + 1), then send one or more Nack control frames requesting retransmission of the missing frame(s).

The destination address for the Nack will be the source address of the received frame. The source address will be this station's MAC address. The destination address of the received frame shall be placed in the original destination address field (NACK_DA) in the LARQ Nack control frame header. The Multiple Retransmission flag shall be set to 0. The [first] missing sequence number shall be placed in the sequence number field. The priority for the NACK control frame shall be the same as the priority for the channel.

If multiple Nack control frames shall be sent, the earliest sequence number shall be sent first.

For each missing sequence number, a Nack retransmission timer shall be started, set to expire at the current time plus Nack Retransmission Interval.

For each missing sequence number, a lost frame timer shall be started, set to expire at the current time plus Maximum Hold Interval.

If the frame is a good data frame and was not delivered to the next layer, then save it.

If the frame is a reminder frame (or an errored data frame), then drop it.

Advance the Current Sequence Number to the sequence number in the received frame.

#### 10.7.5.6 Receiver – Old sequence number

If the sequence number is the same or older than Current Sequence Number, then it shall not generate control frames, although it may itself be dropped, held, or sent up to the next higher layer, possibly causing other held frames to be sent up as well. It may cause the cancellation of a Nack retransmission timer or lost frame timer associated with that sequence number.

If the frame is not a good (e.g., bad CRC) data frame, or its sequence number is older than the oldest missing frame, or it has already been received (this is a duplicate retransmission), or it is a Reminder frame, then drop the frame and skip further processing for this frame.

Cancel the Nack retransmission timer and the lost frame timer for the sequence number.

If the sequence is not the oldest missing sequence number, then save the frame.

If the sequence number is the oldest missing sequence number, then deliver the frame up to the next higher layer. If there is a saved frame for the next sequence number, then deliver in-sequence frames to the layer above until the next sequence number with a missing frame is reached (which may be the next expected sequence number for the channel). The value from the Priority/Flow ID field from the LARQ header for each frame shall be delivered to the next layer along with each associated frame. The method of specifying Priority/Flow ID to the next layer is implementation dependent and outside the scope of this Recommendation.

#### 10.7.5.7 Receiver – Nack Retransmission timer expires

If a Nack retransmission timer expires, then send another Nack control frame for the associated sequence number. The priority for the NACK control frame shall be the same as the priority for the channel. Multiple sequence numbers may be nacked at the same time, if their timers expire at similar times.

The Multiple Retransmission flag shall be set to 1 for Nack control frames sent as a result of retransmission timer expiration.

While there is no explicit limit on the number of Nack control frames sent for a particular sequence number, the Nack timer shall be cancelled if the frame will be received or if the sequence number will be declared lost.

#### 10.7.5.8 Receiver – Lost frame timer expires

The lost frame timer is implementation dependent. Its purpose is to set an upper bound on how long frames will be held before they are sent up when a frame is really lost. If set too long, network resources may be wasted on NACK control frames sent for frames that the sender on the channel would never retransmit. Further, higher layer transport timers may also become involved. The default value of 150 ms is strongly suggested as an upper bound.

Upon expiration, the sequence number shall be declared lost, resulting in the cancellation of the Nack retransmission timer and the lost frame timer for the sequence number. If there is a saved frame for the next sequence number, then send up in-sequence frames until the next sequence number with a missing frame is reached (which may be the next expected sequence number for the channel).

If the lost frame timers for multiple sequence numbers expire at the same time, then the timers are processed in sequence from oldest to newest.

#### 10.7.5.9 Receiver – Forget timer

The forget timer is an implementation-dependent mechanism to allow a receiver to reset the sequence number space of a channel when a received sequence number is not the next expected (Current Sequence Number + 1) and a relatively long interval has expired since the last frame received on the channel. Once expired, a receiver should accept any unusual sequence number as the next expected sequence number, allowing for undetected resets of other stations, disconnection from the network, etc. The definition of "unusual sequence number" is implementation dependent, but generally means any old sequence number or any new sequence number that is not close to the current sequence number, where "close" is 1 or some other small integer. A one-second default is suggested.

#### 10.7.5.10 Receiver – Resource management

In general, the receiver should set upper bounds on the number of held frames per channel and the number of held frames across channels. The bounds may vary based on the Priority/Flow ID of the channel.

Timer intervals may vary based on factors such as the Priority/Flow ID of the channel, or measured intervals for successful retransmissions.

The description above suggests per-sequence number timers. This is for descriptive purposes only, and does not imply any implementation mechanism.

### 10.8 Vendor-specific formats

The following two types (see Tables 10-20 and 10-21) allow vendor-specific extensions which may be reasonably handled by implementations that do not otherwise support them. The short-format vendor-specific format allows short control messages and encapsulation headers, while the long-format subtype allows other extensions that require longer messages.

**Table 10-20/G.9954 – Vendor-specific short frame**

| Field | Length | Meaning |
|---|---|---|
| DA | 6 octets | Destination Address |
| SA | 6 octets | Source Address |
| Ethertype | 2 octets | 0x886c (PNT Link Control Frame) |
| SSType | 1 octet | = SUBTYPE_VENDOR_SHORT (5) |
| SSLength | 1 octet | Number of additional octets in the control header, starting with the SSVersion field and ending with the second (last) octet of the Next Ethertype field. SSLength shall be ≥ 6 for SSVersion 0. |
| SSVersion | 1 octet | = 0 |
| Vendor OUI | 3 octets | An IEEE-assigned Organizationally Unique Identifier |
| Control data | 0-249 octets | Vendor-specific control data |
| Next Ethertype | 2 octets | = next Ethertype if an encapsulation format, or 0 if no encapsulated frame |
| Pad | 0-38 octets | Any value octet |
| FCS | 4 octets | |
| CRC-16 | 2 octets | PNT Frame Check Sequence |

**Table 10-21/G.9954 – Vendor-specific long frame**

| Field | Length | Meaning |
|---|---|---|
| DA | 6 octets | Destination Address |
| SA | 6 octets | Source Address |
| Ethertype | 2 octets | 0x886c (PNT Link Control Frame) |
| LSType | 2 octets | = SUBTYPE_VENDOR_LONG (32769) |
| LSLength | 2 octets | Number of additional octets starting with the LSVersion field and ending with the second (last) octet of the Next Ethertype field. LSLength shall be > 6 for LSVersion 0. |

**Table 10-21/G.9954 – Vendor-specific long frame**

| Field | Length | Meaning |
|---|---|---|
| LSVersion | 1 octet | = 0 |
| Vendor OUI | 3 octets | An IEEE-assigned Organizationally Unique Identifier |
| Control data | 1-65531 octets | Vendor specific data |
| Next Ethertype | 2 octets | = next Ethertype if an encapsulation format, or 0 if no encapsulated frame |
| Pad | 40-0 octets | If needed to make minimum size frame. Should be zero |
| FCS | 4 octets | |
| CRC-16 | 2 octets | PNT Frame Check Sequence |

## 10.9 PNT Certification and Diagnostics Protocol

### 10.9.1 Scope

This protocol is required for G.9954-compliant nodes being submitted for certification testing. Use of this protocol by G.9954 nodes is required.

Devices submitted for PNT Certification testing only need to implement the server portion of the protocol. The same driver implementation should be used for both certification testing and production devices. However, for devices that have stringent resource constraints, the Certification and Diagnostics Protocol may be implemented in a special driver used only for the certification tests.

### 10.9.2 Overview

The PNT Certification and Diagnostics Protocol is designed to provide the required framework for testing of systems providing PNT interfaces. Specifically, it aims to provide a common set of functionality required (equivalent to cert_tool.exe and the UDP functionality of epi_ttcp) for certification testing while minimizing the impact on system design. This protocol is a component of a solution which should provide a control and test interface that enables execution and reporting of a complete certification test case suite, regardless of DUT implementation.

This Recommendation specifies the protocol itself, and does not address details of using the protocol for a specific test or diagnostic function. Such details are dependent on the specific test(s) being performed (e.g., PNT Certification testing vs. network diagnostics) and as such are outside the scope of this Recommendation.

The protocol is designed to be operating system and platform independent, and is intended to support certification testing, with possible extensions to support network-wide diagnostics, system development, and manufacturing and QA testing.

For brevity, we will use the term "cert" to refer to the G.9954 Certification and Diagnostics Protocol.

All cert activity (control and data frames) is restricted to the physical segment under test. There is no support for doing cert through another interface. All control frames received on an interface are only relevant to that interface.

### 10.9.3 Control

One node on the network is the protocol controller, which will be referred to as the "client". The client initiates and coordinates all certification and diagnostics activity. The client portion of the protocol should be enabled on only one node in a network at any time.

All other nodes on the network are "servers". They service requests from the client by adjusting their configuration as directed by the client, or by sourcing and sinking cert data frames as requested by the client. Client nodes should also provide all the functionality of a server. Generally, the server will be implemented within the device driver for PNT nodes, but it may be implemented at a higher layer above any network device assuming that PNT Link Control Frame (LCF) frames can be passed by the server to and from the device driver. In order to minimize the impact on system resources, the server functionality of the cert protocol is intended to be as minimal and straightforward as possible. Cert frames are grouped into two categories: control and data frames. Control frames are used to configure nodes and collect information from nodes. Data frames are used to test transmit and receive capabilities of nodes. Control request frames are only generated by the client. Servers generate replies to the control requests, and generate data frames as directed by the client.

Servers shall reply to control requests within 5 seconds. Servers shall complete any configuration changes (e.g., PNT mode changes) initiated by the control request within 5 seconds after receipt of the control request.

Devices submitted for PNT Certification testing using this protocol shall implement the server portion of the protocol. Implementation of the client portion of the protocol is not required. Cert frames shall not be bridged by any node.

Control frames should be sent at link layer (LL) priority 7. Data frames shall be sent at the LL priority/Flow ID specified by the client when initiating the data transmission. If any encapsulating protocols (e.g., LARQ) are enabled on a node, the data frames shall be sent with the enabled encapsulation(s) to facilitate testing of the protocol implementation(s). Control frames may be encapsulated. Cert clients and servers shall be able to de-encapsulate cert control frames to the same extent that they are required to de-encapsulate data frames. PNT nodes shall be capable of removing one encapsulating Short-Format Link Control Frame header from cert control frames.

### 10.9.4 Frame format

Cert frames use the basic PNT Link Layer Control Frame (LCF) format defined in "Interface Specification for PNT Technology Link Layer Protocols". A single long-subtype frame format is defined with a common header structure used with all cert frames and a variable number of command or data segments (see Table 10-22):

**Table 10-22/G.9954 – Certification and diagnostics frame format**

| Field | Length | Meaning |
|---|---|---|
| DA | 6 octets | Destination Address |
| SA | 6 octets | Source Address |
| Ethertype | 2 octets | 0x886c (PNT Link Control Frame) |
| LSType | 2 octets | = SUBTYPE_CERT (32770) |
| LSLength | 2 octets | Number of additional octets in the control header, starting with the LSVersion field and ending with the second (last) octet of the Next Ethertype field. Minimum is 6 for LSVersion 0 |
| LSVersion | 1 octet | = 0 |
| OpCode | 1 octet | Command segment set used in this frame |
| Reserved | 4 octets | |
| Cert_Seq | 2 octets | Frame sequence number |
| CommandData | 0-1486 octets | Command data, may be empty, or contain one or more Command Segments, or one Data Segment |

**Table 10-22/G.9954 – Certification and diagnostics frame format**

| Field | Length | Meaning |
|---|---|---|
| Next Ethertype | 2 octets | = 0 |
| Pad | 40-0 octets | Should be zero |
| FCS | 4 octets | |
| CRC-16 | 2 octets | PNT Frame Check Sequence |

The command segments use the format shown in Table 10-23:

**Table 10-23/G.9954 – Command segment format**

| Field | Length | Meaning |
|---|---|---|
| CSType | 2 octets | The type of the command segment |
| CSLength | 2 octets | Number of octets in the CSPayload field. Valid values are nominally 0-1482. However, for some CSType values the CSLength field is fixed.<br>The high 3 bits are reserved in version 0; they shall be sent as 0 and ignored on reception. |
| CSPayload | 0-1482 octets | Command-specific information. May be empty |
| CSPad | 0-3 octets | If present, shall be sent as 0, ignored on reception. Aligns subsequent Command Segments on 32-bit boundaries. Shall be present if CSLength is not a multiple of 4 |

Data segments use the format shown in Table 10-24:

**Table 10-24/G.9954 – Data segment format**

| Field | Length | Meaning |
|---|---|---|
| DSType | 2 octets | The type of the data segment. |
| DSLength | 2 octets | Number of octets in the DSPayload field. Valid values are nominally 1-1482.<br>The high 3 bits are reserved in version 0; they shall be sent as 0 and ignored on reception. |
| DSPayload | 1-1482 octets | Data. |

Replies from a server can span multiple frames, but individual command segments shall not extend across frame boundaries.

When multiple command segments are present in a frame, they shall be sent in order by ascending tag value.

All Command segments shall be aligned on 4-byte boundaries. All command segments shall be padded to a multiple of 4 bytes. Data segments shall not be padded, and shall not be combined with command segments.

### 10.9.5 Opcodes

Server nodes generate the opcodes in Table 10-25:

**Table 10-25/G.9954 – Server node opcodes**

| Mnemonic | Opcode |
|---|---|
| OK | 0x00 |
| ERROR | 0x01 |
| TESTDATA | 0x02 |
| SAMPLEDATA | 0x03 |

Client nodes generate the opcodes in Table 10-26:

**Table 10-26/G.9954 – Client node opcodes**

| Mnemonic | Opcode |
|---|---|
| ENABLECERT | 0x08 |
| DISABLECERT | 0x09 |
| CONFIGNODE | 0x10 |
| CONFIGSEND | 0x11 |
| STARTSEND | 0x12 |
| STOPSEND | 0x13 |
| ECHOREQUEST | 0x14 |
| CONFIGRECV | 0x15 |
| STOPRECV | 0x16 |
| REPORTSTATS | 0x17 |
| REPORTCONFIG | 0x18 |
| RESETSTATS | 0x19 |
| REPORTNODE | 0x20 |
| STARTSAMPLE | 0x30 |
| VENDOR | 0x40 |

### 10.9.6 Command segments

Command segments are listed in groups (see Table 10-27), with the opcode(s) that use them preceding each group.

**Table 10-27/G.9954 – Command segment groups**

| Mnemonic | CSType | CSLen | CSPayload values | Description |
|---|---|---|---|---|
| Opcode: ERROR | | | | |
| ERRORCODE | 0x0001 | 1 | 1-8 | An index indicating the error from the list:<br>1 UNK<br>2 UNSUP_OP<br>3 INVALID_PARAM<br>4 UNSUP_CMDSEG<br>5 UNSUP_DGEN<br>6 INVALID_SEQ<br>7 INVALID_FRAME<br>8 INVALID_OP |
| Opcodes:<br>OK(REPORTCONFIG)<br>OK(REPORTSTATS)<br>OK(REPORTNODE) | | | | |
| INFOREPLY | 0x0002 | 2 | Two 8-bit values | Number of reply frames – 1, plus Index of current frame (starting with 0). |
| Opcodes:<br>STARTSEND<br>STOPSEND<br>STOPRECV | | | | |
| REFSEQ | 0x0005 | 2 | Any | REFSEQ value contains the Cert_Seq value from a previous command. |
| Opcode: VENDOR | | | | |
| OUI | 0x0023 | 3 | IEEE OUI | Vendor commands are sent with this command segment first. |
| Opcodes:<br>CONFIG_NODE<br>OK (in response to REPORTCONFIG) | | | | |
| TXPE | 0x0010 | 1 | 1-7<br>9-15 (optional)<br>255 (default) | Fixed PE, rate negotiation disabled<br>Fixed PE, rate negotiation disabled<br>Rate negotiation enabled |
| TXPRI | 0x0011 | 1 | 0-7<br>255 (default) | Fixed transmit PHY priority<br>Use LL priority, negotiate priority map via CSA |
| LINKINT | 0x0012 | 1 | 0<br>1 (default) | Link Integrity disabled<br>Link Integrity enabled |

**Table 10-27/G.9954 – Command segment groups**

| Mnemonic | CSType | CSLen | CSPayload values | Description |
|---|---|---|---|---|
| TXMODE | 0x0013 | 1 | 0<br>1 (default)<br>2 | Disable all transmissions<br>Enable all transmissions<br>Enable only PNT Link Control Frame transmissions |
| HPNAMODE | 0x0016 | 1 | 0 (default)<br><br>1<br>2<br><br>3 (optional)<br>4<br>5 (optional)<br>6<br>7 | Automatic switching between modes<br>Reserved for legacy usage<br>Forces G.9951/2 mode (Spectral Mask #1)<br>Reserved for legacy usage<br>Reserved for legacy usage<br>Reserved for legacy usage<br>Force Spectral Mask #2 mode<br>Force Spectral Mask #3 mode |
| LARQ (optional) | 0x0020 | 1 | 0<br><br>1 | LARQ disabled (but headers stripped).<br>LARQ enabled |
| CSA (optional) | 0x0021 | 1 | 0<br>1 | CSA disabled.<br>CSA enabled. |
| CSAHPNAMODE (optional) | 0x0022 | 1 | 0 (default)<br><br>1<br>2<br><br>3 | Do not set any mode config flags in CSA messages<br>Reserved for legacy usage<br>Set ConfigG.9951/2 flag in CSA messages.<br>Reserved for legacy usage. |
| Opcodes: STARTSAMPLE | | | | |
| SAMPLE | 0x0030 | 14 | MAC address | Octet 0-5: SA of channel |
| | | | MAC address | Octet 6-11: DA of channel |
| | | | 0 = none<br>1 = GAP<br>2 = PREAMBLE | Octet 12: Test type |
| | | | 0 | Octet 13: Reserved – Shall be set to zero by the transmitter and ignored by the receiver. |
| Opcodes: CONFIGSEND CONFIGRECV | | | | |
| DGEN_TYPE | 0x0084 | 1 | 1,2 | Data generator to use for the data segment of the frames. See 10.9.17. |
| DGEN_DATA | 0x0085 | 4 | Any | Initialization value for data generator. See 10.9.17. |
| LENGTH | 0x0086 | 2 | 1-1482 | Length of the data segment of the frames to be sent. |

**Table 10-27/G.9954 – Command segment groups**

| Mnemonic | CSType | CSLen | CSPayload values | Description |
|---|---|---|---|---|
| SA | 0x0081 | 6 | Unicast MAC address | MAC address of the node that will be the source of the data frames (generally the MAC address of the recipient of the CONFIGSEND request). |
| DA | 0x0083 | 6 | Any MAC address | MAC address of the node(s) that will be the recipient(s) of the data frames. A total of ten DA segments may be present, and must be supported. |
| Opcode: CONFIGSEND | | | | |
| NPKTS | 0x0087 | 4 | Any<br>(default = 0) | Total number of packets to send.<br>0 means send frames continuously until a STOPSEND request is received |
| BURST_INT | 0x0088 | 2 | Any<br>(default = 0) | Interval between start of bursts in milliseconds. 0 means send frames without any pacing |
| BURST_NPKTS | 0x0089 | 2 | !=0<br>(default = 1) | Number of packets to send per burst |
| NUMACKS | 0x008a | 1 | !=0<br>(default = 1) | Number of ACK and EOT frames to send. (See 10.9.10.2.) |
| TXPE_TEST | 0x008b | 1 | 1-7<br>9-15 (optional)<br>255 (default) | Fixed PE, rate negotiation disabled<br>Fixed PE, rate negotiation disabled<br>Rate negotiation enabled<br>Applies only to test frames being generated by the server |
| TXPRI_TEST | 0x008c | 1 | 0-7<br>255 (default) | Fixed transmit PHY priority<br>Use LL priority, negotiate priority map via CSA<br>Applies only to test frames being generated by the server |
| Opcode:<br>OK (in response to REPORT_STATS) | | | | |
| RECV_NPKTS | 0x0105 | 4 | Any | Total number of data frames received without errors, not including EOT frames |
| RECV_NBYTES | 0x0106 | 4 | Any | Total number of data bytes received without errors |
| RECV_SEQ_MISS | 0x0107 | 4 | Any | Number of missing data frames detected via gaps in sequence numbers |
| RECV_SEQ_ERR | 0x0108 | 4 | Any | Number of data frames received with unexpected sequence numbers |

**Table 10-27/G.9954 – Command segment groups**

| Mnemonic | CSType | CSLen | CSPayload values | Description |
|---|---|---|---|---|
| RECV_DATA_ERR | 0x0109 | 4 | Any | Number of data frames received with detected data corruption |
| RECV_FCS_ERR | 0x010c | 4 | Any | Number of frames received with FCS errors |
| RECV_HDR_ERR | | 4 | Any | Number of frames received with detected header errors |
| RECV_ERR | 0x010a | 4 | Any | Number of frames with other recv errors |
| RECV_ELAPSED_TIME | 0x010b | 4 | Any | Receive test elapsed time in ms |
| XMT_NPKTS | 0x0101 | 4 | Any | Total number of data frames sent without errors reported by lower layers (e.g., excessive collisions), not including EOT frames |
| XMT_NBYTES | 0x0102 | 4 | Any | Total number of data bytes sent without errors |
| XMT_NERRS | 0x0103 | 4 | Any | Number of transmit errors reported by lower layers that resulted in lost frames (e.g., excessive collisions) |
| XMT_ELAPSED_TIME | 0x0104 | 4 | Any | Transmit elapsed time in ms |
| Opcode:<br>OK (in response to REPORTNODE) | | | | |
| PRIMARY_ID | 0x8301 | 4 | Any | Primary Vendor/Device ID |
| SUBSYSTEM_ID | 0x8302 | 4 | Any | Subsystem Vendor/Device ID |
| MAC_ADDRESS | 0x8303 | 6 | Any | IEEE 48-bit MAC address |
| SERIAL_NUM | 0x8304 | ≤16 | ASCII | |

**Table 10-27/G.9954 – Command segment groups**

| Mnemonic | CSType | CSLen | CSPayload values | Description |
|---|---|---|---|---|
| DEVICE_TYPE | 0x8305 | 1 | 0-24 | An index indicating the device type:<br>0 Other<br>1 PCI NIC (includes miniPCI, Cardbus)<br>2 USB NIC<br>3 Cable Modem Bridge<br>4 DSL Modem Bridge<br>5 Broadband Wireless Bridge<br>6 V90 Bridge<br>7 Stand-alone Bridge<br>8 Cable Modem Router<br>9 DSL Modem Router<br>10 Broadband Wireless Router<br>11 V90 Router<br>12 Standalone Router<br>13 Audio Device<br>14 Video Device<br>15 Disk Device<br>16 CD/DVD Device<br>17 Backup Device<br>18 Digital Cable Set-top<br>19 Digital Satellite Set-top<br>20 Printer<br>21 Print Server<br>22 Scanner<br>23 FAX<br>24 Phone |
| VEND_NAME | 0x8306 | ≤32 | ASCII | |
| VEND_DRIVER | 0x8307 | ≤16 | ASCII | |
| VEND_DATE | 0x8308 | 4 | TBD | |
| MANUF_DATE | 0x8309 | 4 | TBD | |
| TIMER_GRAN | 0x830a | 2 | 1-1000 | Timer resolution in ms |

### 10.9.7 Data segments

Data segments are listed in groups (see Table 10-28), with the opcode(s) that use them preceding each group.

**Table 10-28/G.9954 – Data segment groups**

| Mnemonic | CSType | CSLen | CSPayload values | Description |
|---|---|---|---|---|
| Opcodes: TESTDATA ECHOREQUEST<br>OK (in response to ECHOREQUEST) | | | | |
| DATA | 0x8108 | 1-1482 | Any | Data |
| Opcode: TESTDATA | | | | |
| EOT | 0x8109 | 0 | N/A | End of Transmission: Marks the end of the server's data transmission |
| Opcode: SAMPLEDATA | | | | |
| SAMPLES | 0x8133 | 1-1482 | MAC address | Octet 0-5: Source address of channel |
| | | | 0-65535 | Octet 6-7: Total number of samples in test |
| | | | 0-65535 | Octet 8-9: Index of first sample in this segment |
| | | | 0 None<br>1 GAP<br>2 PREAMBLE | Octet 10: The test type (from CSPayload of command segment) |
| | | | 0 | Octet 11: Reserved for future use. Shall be set to zero by the transmitter and ignored by the receiver. |
| | | | Samples… | Octet 12 to (DSLength-13): Signed 16-bit samples |

### 10.9.8 Server opcode usage

#### 10.9.8.1 OK

Opcode OK messages are generated in response to control requests that are successfully completed. Opcode OK messages contain variable number of command segments, depending on the control request. Opcode OK messages with zero command segments are referred to as "empty OK" messages.

The Cert_Seq field in the OK message shall be set to the value of the Cert_Seq field from the control request.

If multiple OK messages are being generated in response to a single command request, the INFOREPLY command segment shall be the first segment in each reply frame. An INFOREPLY command segment may be included as the first command segment when a single OK message is being generated.

#### 10.9.8.2 OK ERROR

Opcode ERROR messages are generated in response to control requests which are malformed, not understood, or could not be completed successfully. The Cert_Seq field in the ERROR message shall be set to the value of the Cert_Seq field from the control request. Opcode ERROR messages shall contain one or two command segments. The first command segment shall have CSType = ERRORCODE. The second command segment, if present, shall be an ERRORPOINTER command

segment with CSPayload containing the first four octets (CSType and CSLength) from the first command segment that caused the problem, if it can be identified.

#### 10.9.8.3 TESTDATA

Opcode TESTDATA frames are used to measure performance (e.g., frame error rate) or implementation (e.g., in-order delivery of LARQ encapsulated frames) characteristics of the nodes being tested, and are typically sent between two servers. The Cert_Seq field in the TESTDATA messages typically starts at 0 for each test, and increases by one for each subsequent TESTDATA frame sent as part of that test.

Opcode TESTDATA messages shall contain a single data segment of with DSType = DATA or command segment with CSType = EOT.

#### 10.9.8.4 SAMPLEDATA

Opcode SAMPLEDATA frames are used to support a spectral analysis of a PNT channel from server A to server B as seen by server B. Upon receiving STARTSAMPLE command, the source of the tested channel shall send a Link-Layer Link Integrity message to the destination of the channel. The destination of the channel shall send SAMPLES data segment(s) to the server containing 32 symbols worth of samples using its native sample rate. If the samples span more than one data segment, then the segments should be sent in an ascending order of sample index.

When the test type is PREAMBLE, the samples shall represent symbols 25 to 56 of the preamble for the frame received from the source of the channel.

When the test type is GAP, the samples shall represent a period in the inter-frame-gap that starts 8 microseconds after the reception of the frame.

### 10.9.9 Client opcode usage

#### 10.9.9.1 ENABLECERT

At startup, or after receipt of a DISABLECERT request, servers shall be in "cert disabled" mode. While in "cert disabled" mode, the node shall silently ignore all received cert frames except DISABLECERT and ENABLECERT requests until an error-free ENABLECERT request has been received. After receipt of an ENABLECERT request, the node shall check the format of the received frame. If no errors are detected, the node shall switch to (or remain in) "cert enabled" mode and reply with an empty OK message. If an error in the frame format is detected, the node shall reply with an ERROR message and shall not switch modes.

#### 10.9.9.2 DISABLECERT

After receipt of a DISABLECERT request, servers shall check the format of the received frame. If no errors are detected, the node shall reply with an empty OK message, switch to (or remain in) "cert disabled" mode, and then silently ignore all subsequent received cert frames except DISABLECERT and ENABLECERT requests. If an error in the frame format is detected, the node shall reply with an ERROR message and shall not switch modes.

#### 10.9.9.3 CONFIGNODE

Opcode CONFIGNODE messages may contain exactly one of the following command segments:

- TXPE;
- TXPRI;
- LINKINT;
- TXMODE;
- HPNAMODE;
- LARQ;

- CSA;
- CSAHPNAMODE.

All servers shall support the TXPRI, LINKINT, and TXMODE command segments. All servers shall support TXPE settings 1-7 and 255. Servers shall support TXPE settings 9-15 if and only if they are capable of transmitting 4-Mbaud payloads. All servers shall support HPNAMODE settings 0, 2, 6 and 7. Servers shall support the LARQ command segment if and only if they implement the LARQ protocol. Servers shall support the CSA and CSAHPNAMODE command segments if and only if they implement the CSA protocol.

If a server receives a CONFIGNODE request with an unsupported or invalid command segment, it shall reply with an ERROR message. Otherwise, it shall reply with an empty OK message.

#### 10.9.9.4 CONFIGSEND

These command segments shall be provided in a CONFIGSEND request, in the order listed:

- DGEN_TYPE;
- DGEN_DATA;
- LENGTH;
- SA;
- DA.

DA is the only CSType in a CONFIGSEND request that may be repeated, and if repeated, all DA segments shall be contiguous. Implementations shall support at least ten DA command segments in a CONFIGSEND request. CONFIGSEND command segments shall only be sent to unicast addresses.

The traffic generator is responsible for generating the data in the frames, the size of the frames, and the distribution of frames in the case of multiple DAs. The most commonly used generator is fixed data, fixed length frames, round robin distribution to all DAs.

The following command segments are optional in a CONFIGSEND request, but if present, all shall be sent in the order listed:

- NPKTS;
- BURST_INT;
- BURST_NPKTS;
- NUMACKS;
- TXPE_TEST;
- TXPRI_TEST.

If the server cannot provide the resolution implied by BURST_INT, then the value shall be rounded up to the closest value which the server can provide.

If BURST_INT is not specified or is 0, then the data sending node shall generate frames as fast as possible without dropping frames on the transmit side.

If NPKTS is not specified or is 0, then the data-sending node shall generate data frames until a STOPSEND request is received.

The receiving node shall reply with an ERROR message if any unsupported parameters (or unsupported values for supported parameters) are included in the CONFIGSEND request, if the receiving node is already in the process of sending cert data frames from a previous CONFIGSEND/STARTSEND set of requests, if more than one CONFIGSEND is received before a STARTSEND request is received, or if the SA in the CONFIGSEND request is not the receiving node's MAC address. Otherwise, the receiving node shall reset the transmit counters listed in

10.9.11, set any optional parameters not included in the CONFIGSEND request to their default values, and reply with an empty OK message.

#### 10.9.9.5 STARTSEND

STARTSEND requests contain one or more command segments of CSType = REFSEQ. Each REFSEQ value matches the Cert_Seq value of a CONFIGSEND request that was previously issued. Receiving nodes shall follow the protocol defined in 10.9.10.2.

#### 10.9.9.6 STOPSEND

STOPSEND requests include one or more command segments of CSType = REFSEQ. Each REFSEQ value matches the Cert_Seq value from a CONFIGSEND request that created a data stream. When a server receives a STOPSEND request, it compares the Cert_Seq value(s) in the request to the Cert_Seq value from the last CONFIGSEND request it received. If there is a match, the server shall reply with a single OK message, containing one command segment of CSType = REFSEQ with the Cert_Seq value that matched. If a STOPSEND request is received while data frames are being sent, the transmitting node shall stop sending data frames. If there is no match, or if the node has not received any CONFIGSEND requests, then it shall silently ignore the request.

#### 10.9.9.7 ECHOREQUEST

ECHOREQUEST frames contain a single data segment of DSType = DATA. The client fills the DSPayload field with the data it wishes to get echoed back (from 1 to 1482 bytes), and sets the DSLength field appropriately. The receiver shall reply with an OK message containing a copy of the data segment from the ECHOREQUEST command.

#### 10.9.9.8 CONFIGRECV

These command segments shall be provided in a CONFIGRECV request, in the order listed:

- DGEN_TYPE;
- DGEN_DATA;
- LENGTH;
- SA;
- DA.

DA is the only CSType in a CONFIGRECV request that may be repeated, and if repeated, all DA segments shall be contiguous. CONFIGRECV command segments shall only be sent to unicast addresses.

The receiving node shall reply with a single ERROR message if any unsupported parameters (or unsupported values for supported parameters) are included in the CONFIGRECV request, or if its MAC address does not appear in any of the DA command segments. Otherwise, the receiving node shall reset the receive counters listed in 10.9.11, set any optional parameters not included in the CONFIGRECV request to their default values, and reply with an empty OK message.

#### 10.9.9.9 STOPRECV

STOPRECV requests include one or more command segments of CSType = REFSEQ. Each REFSEQ value matches the Cert_Seq value from a CONFIGRECV request that created a data stream. When a server receives a STOPRECV request, it compares the Cert_Seq value(s) in the request to the Cert_Seq value from the last CONFIGRECV request it received. If there is a match, the server shall immediately compute the elapsed time from the start of the test or, if no data frames have been received, set the elapsed time to zero, and reply with a single OK message, containing one command segment of CSType = REFSEQ with the Cert_Seq value that matched. Any subsequent data frames received shall be ignored. If there is no match, or if the node has not received any CONFIGRECV requests, then it shall silently ignore the request.

#### 10.9.9.10 REPORTSTATS

The receiver shall reply with an OK message containing the counters listed in 10.9.11, in the order listed in that clause. The counters shall not be reset after being reported, in case the reply is lost and the client needs to repeat the REPORTSTATS command. The reply message shall start with an INFOREPLY command segment, followed by command segments for each of the required counters.

#### 10.9.9.11 REPORTCONFIG

The receiver shall reply with an OK message containing the current settings for the configuration parameters listed in 10.9.9.3. The reply message shall start with an INFOREPLY command segment, followed by command segments for each of the required parameters. The command segments shall be sent in the order listed in 10.9.9.3. The first five configuration parameters shall be reported, while the last three, LARQ, CSA, and CSAHPNAMODE, shall be reported only if supported.

#### 10.9.9.12 RESETSTATS

The receiver shall reset all the counters listed in 10.9.11 and reply with an empty OK message.

#### 10.9.9.13 REPORTNODE

The receiver shall reply with an OK message containing fixed information pertaining to the node, such as identifiers, software/hardware versions, etc. The reply frames shall each begin with an INFOREPLY command segment, followed by command segments from the following list, sent in the order listed:

- PRIMARY_ID;
- SUBSYSTEM_ID;
- MAC_ADDRESS;
- SERIAL_NUM;
- DEVICE_TYPE;
- VEND_NAME;
- VEND_DRIVER;
- VEND_DATE;
- MANUF_DATE;
- TIMER_GRAN.

#### 10.9.9.14 STARTSAMPLE

The client shall initiate sampling of the channel by sending a SAMPLE command segment. The DA of the Certification and Diagnostics frame shall be BROADCAST. The client shall then wait for all the "SAMPLES" data segments to arrive. The application should use a proper timeout in case the server(s) do not reply.

#### 10.9.9.15 VENDOR

This opcode allows vendors to implement a private set of functions. The first command segment shall be CSType = OUI with CSPayload set to the vendor's OUI. A node receiving a vendor-specific command request with an OUI that does not match an OUI it understands shall return an INVALID_PARAM error message. The behaviour of nodes that receive a vendor-specific command request with a matching OUI is at the discretion of the vendor, and is outside the scope of this Recommendation.

### 10.9.10 Control Request Protocol

#### 10.9.10.1 General control requests

All control requests other than STARTSEND and VENDOR follow a trivial protocol: The client sends a single-frame request, and the server replies with one or more frames – all control frames sent by the client are explicitly "acked" with either OK or ERROR or SAMPLEDATA in case of STARTSAMPLE. For most cases, a single frame is generated. Each control frame generated by the client shall be sent with a monotonically increasing (ignoring rollover) value for Cert_Seq. The Cert_Seq field in the acknowledgement frames from the server nodes use the Cert_Seq value from the control request to ensure that the client can properly identify which request is being acked. The client shall be responsible for dealing with unacknowledged requests, e.g., by resending the request after some timeout, with a possible delay between attempts. Failure to receive an ack can mean that either the original request frame was lost, or the ack was lost. For all currently defined requests except STARTSEND, there is no negative impact to resending a request. The timeout value used by the client is dependent on the specific request being issued. For config commands, a timeout of 50 ms should be used. The client behaviour if repeated failures are encountered is dependent on the goals of the client (certification testing vs. network diagnostics) and is not specified here.

In the case of a REPORTSTATS or REPORTCONFIG request, some number (≥1) of reply frames are generated by the server. The first command segment of all reply frames sent in response to REPORTSTATS, REPORTCONFIG, and REPORTNODE requests shall be an INFOREPLY command segment indicating the total number of frames to be sent and the relative number of the current frame.

Subsequent command segments contain the data being returned by the server.

All reply frames are sent with Cert_Seq value set to the Cert_Seq from the client's request. The client shall be responsible for ensuring that all frames have been received and reissuing the request if any frames will be lost.

#### 10.9.10.2 Protocol for STARTSEND control requests

In order to provide an uninterrupted flow of data frames during a test, a somewhat different protocol shall be used for STARTSEND requests. After issuing the appropriate CONFIGRECV and CONFIGSEND requests to configure all nodes, the client issues a STARTSEND request with a list of control segments of type REFSEQ each containing the Cert_Seq for a previous CONFIGSEND request. Any node expecting a STARTSEND request (i.e., one which has received a CONFIGSEND but has not yet received a STARTSEND request) which receives the STARTSEND request looks through the list of REFSEQ control segments in the STARTSEND request for a Cert_Seq value which matches the sequence number from the CONFIGSEND request. If no match is found, the server silently ignores the STARTSEND request. If a match is found, the node sends NUMACKS OK control replies to the client with a REFSEQ control segment containing the Cert_Seq value of the CONFIGSEND request. The server then sends the requested data frames to the destination address(es). The Cert_Seq field in the data frames starts at zero, and increases by one (modulo 2^16) for each data frame sent. After all data frames have been sent, the server then sends NUMACKS data frames with command segment type EOT, with CSValue set to the sequence number of the CONFIGSEND request, to each of the destination addresses. Upon receipt of the EOT frame, the destination node(s) measure the elapsed time of the data send and discard any data frames received after the EOT. The EOT frame shall not be counted in the receive statistics. A timeline of a typical data test is shown in Figure 10-6.

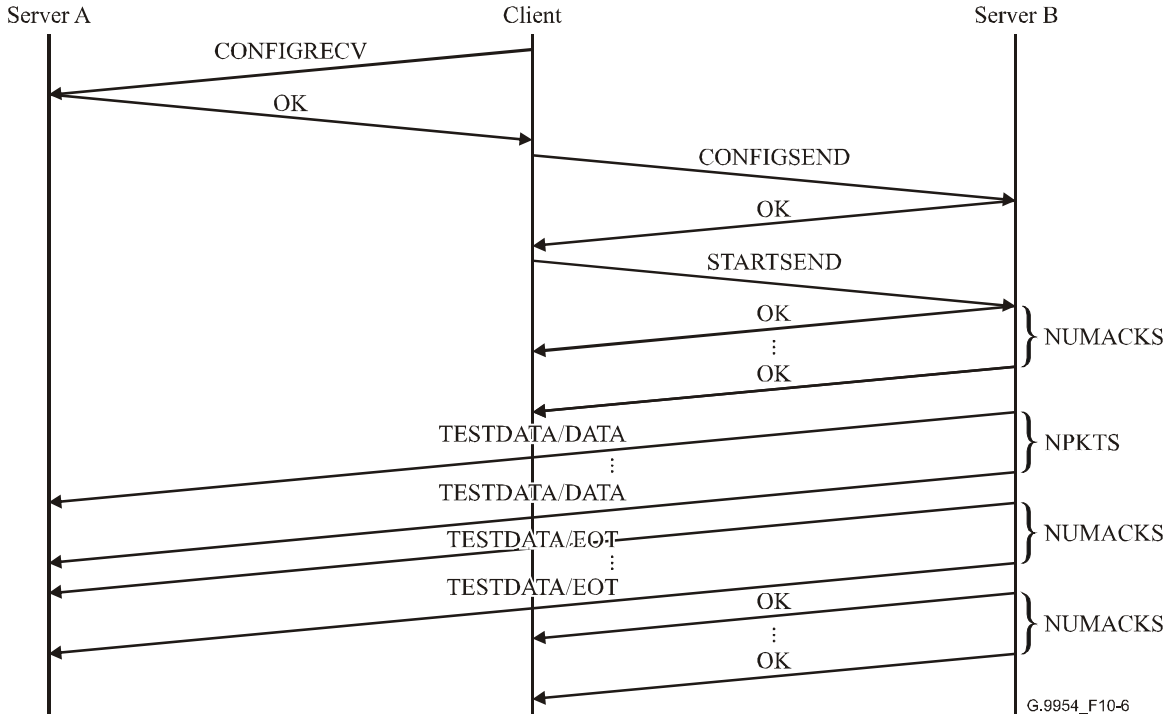


**Figure 10-6/G.9954 – Protocol timeline for data test**

The server then sends NUMACKS OK control reply frames with command segment type EOT, with CSValue set to the sequence number of the CONFIGSEND request, to the client.

If no transmit priority has been configured, then the data frames are sent at the default LL priority 0, and the control reply frames to the client are sent at LL priority 7. The server should ensure that all data frames (including the EOT frames) have been sent on the wire before the control reply frames are sent to the client.

If a server receives a duplicate STARTSEND request for a given CONFIGSEND request (indicating that the client did not receive any of the initial NUMACKS OK control replies), the server shall return an ERROR frame to the client. The client shall be responsible for issuing any necessary STOPSEND requests, reconfiguring the nodes as necessary and restarting the test.

For STARTSEND requests, the timeout which the client should use while looking for the initial acks is 50 ms. The timeout for the final acks (those sent back to the client after the data frames containing EOT segments have been sent) needs to be calculated based on the amount of data being transmitted and the worst-case throughput for the test.

#### 10.9.10.3 Protocol for VENDOR control requests

The protocol for VENDOR control requests is at the discretion of the vendor, and is outside the scope of this Recommendation.

### 10.9.11 Stats

#### 10.9.11.1 Receive counters

The following counters shall be maintained by a server receiving data frames, and reported in response to a REPORTSTATS command:

- RECV_NPKTS;
- RECV_NBYTES;
- RECV_SEQ_MISS;

- RECV_SEQ_ERR;
- RECV_DATA_ERR;
- RECV_FCS_ERR;
- RECV_HDR_ERR;
- RECV_ERR;
- RECV_ELAPSED_TIME.

#### 10.9.11.2 Transmit counters

The following counters shall be maintained by a server sending data frames, and reported in response to a REPORTSTATS command:

- XMT_NPKTS;
- XMT_NBYTES;
- XMT_NERRS;
- XMT_ELAPSED_TIME.

All counters shall be maintained and reported as 32 bits.

Elapsed time shall be measured from transmit or receive of the first data frame until transmit or receive of the first EOT frame.

### 10.9.12 Receiver processing of control frames

Frames with HCS, FCS or CRC-16 errors are not used by cert. Since some implementations may exist as a separate layer above the device driver, there is no guarantee across implementations that frames with these errors will reach the cert layer. Thus, for consistency, all cert implementations shall ignore any frames received with any of these errors.

### 10.9.13 Receiver processing of data frames

For each received data frame:

- If DGEN_TYPE and DGEN_DATA were specified in the CONFIGRECV request, the receiver generates a local copy of the packet using the data generator and compares it to the received packet data. If the data fails to match, the receiver increments recv_data_err. If no errors are detected, the receiver increments recv_npkts.
- The receiver tracks the sequence number of the received frames and increments RECV_SEQ_MISS for any frames that have been missed (as evidenced by gaps in the sequence numbers) and increments RECV_SEQ_ERR for any frames received out of sequence.

The following logic shall be used to increment recv_seq_miss and recv_seq_err:

```
if (((received_seq - expected seq) & 2^15) != 0) recv_seq_err++;
else {
     recv_npkts++;
     if (received_seq == expected_seq) expected_seq = (expected_seq + 1) %
     2^16;
     else {
          if (received_seq > expected_seq) recv_seq_miss += (received_seq -
          expected_seq);
          else recv_seq_miss += (2^16 + received_seq - expected_seq);
          expected_seq = (received_seq + 1) % 2^16;
     }
}
```

Duplicate frames will also increment recv_seq_err.

### 10.9.14 General requirements

Server nodes should be capable of sourcing and sinking data frames simultaneously, but are not required to do so. Servers shall be able to handle receipt and processing of control frames while sending data frames. This version of the protocol does not specify support for simultaneously generating multiple data streams or simultaneously receiving and validating multiple data streams.

### 10.9.15 Timing

The resolution on all timing (timestamps and sending intervals) should be 10 ms, and it shall not be more than 50 ms. Jitter requirement shall be ±10% of the provided resolution.

### 10.9.16 Error codes

The error codes in Table 10-29 have been defined:

**Table 10-29/G.9954 – Error codes**

| Mnemonic | Value |
|---|---|
| UNK | 1 |
| UNSUP_OP | 2 |
| INVALID_PARAM | 3 |
| UNSUP_CMDSEG | 4 |
| UNSUP_DGEN | 5 |
| INVALID_SEQ | 6 |
| INVALID_FRAME | 7 |
| INVALID_OP | 8 |

### 10.9.17 Data generators

#### 10.9.17.1 DGEN_TYPE = 1

The 4 bytes of DGEN_DATA specified in the CONFIGSEND request are replicated, as a group, to fill the length of the payload. If the payload length is not a multiple of 4, the remaining bytes are filled with the portion of DGEN_DATA that fits. E.g., if DGEN_DATA = 0x01020304 and the payload length is 11, then the payload shall be filled with 0x01020304010203040102030401020304010203.

If the number of destination addresses is greater than one, then the generated frames are multiplexed to the destination nodes in the order they were listed in the CONFIGSEND request.

#### 10.9.17.2 DGEN_TYPE = 2

The least significant byte of DGEN_DATA shall be used to initialize an 8-bit counter. Payload bytes shall be sequentially filled with the value of the counter, and the counter shall be incremented by one per payload byte. E.g., if DGEN_DATA = 0xf9 and the payload length is 11, then the payload shall be filled with 0xf9fafbfcfdfeff00010203. If the number of destination addresses is greater than one, then the generated frames are multiplexed to the destination nodes in the order they were listed in the CONFIGSEND request. The three most significant bytes shall be sent as zero and ignored on receipt.

## 10.10 Link-layer framing extensions

This clause of the link-layer specification describes how extensions to frame formats are accomplished.

In addition, two extensions for CSA control frames to support the use of optional and/or extended features between compatible stations are defined. The first extension is a list of optional LCP frame

subtypes supported by the implementation (beyond the four basic version PNT types). New frame types, such as one for a Reed-Solomon encoded frame, would be announced by stations that implement them, allowing for simple pairwise "negotiation" of support for optional types. The second extension is a standard format for announcing parameters associated with an extended feature.

Finally, this clause adds some additional rules governing the design and usage of new/revised LCP protocols, including some more concrete guidelines on LCP header lengths and alignment restrictions.

### 10.10.1 Definitions

**10.10.1.1 embed**: Place data, typically an Ethernet/802.3 frame payload, within the structure defined for an LCP subtype header, possibly encoded, in a manner that requires understanding of the structure to extract the original payload (i.e., the original payload becomes part of the LCP header).

**10.10.1.2 embedded payload**: The data encoded within an embedding header, typically the payload of an Ethernet/802.3 frame starting with the Type/Length field.

**10.10.1.3 embedding header**: Header that contains an embedded payload, for which the header's function must be understood to make use of the enclosed data.

**10.10.1.4 encapsulating header**: A header that can be removed without further processing (e.g., a LARQ header), leaving something useful, typically an Ethernet/802.3 frame payload. An encapsulating header has a non-zero Next Ethertype field.

**10.10.1.5 Encapsulate**: To insert an LCP header into a frame, prior to the original Type/Length field, without modifying the rest of the frame. Removal of the header restores the frame to its original state (i.e., the original payload follows the LCP header).

**10.10.1.6 Tag Length Value (TLV)**: A type of structure consisting of an assigned identifier, the Tag, followed by a Length field specifying the size of the data to follow, followed by the Value (data) itself.

### 10.10.2 Extension mechanism

Extensions to existing frame formats shall be added using Tag-Length-Value (TLV) encoding, with tags assigned by PNT. The TLV format has short and long versions. The short format has an 8-bit tag and an 8-bit length field, while the long-format has a 16-bit tag and a 16-bit length. The short format uses tag values 1-127, and the long format uses tag values 32768-65535, with most significant bit of the most significant octet of the Tag field distinguishing the two formats.

Tags values are assigned independently for each LCP SSType or LSType from the full range of values (i.e., the ranges are overlapping). The tag value 0x00 is explicitly reserved as a pad value, the use of which is described below.

When TLV blocks are added, they shall precede the Next Ethertype field and follow all other non-TLV-encoded fields. The definition of new TLV extensions for a particular subtype does not automatically force the assignment of a new version for SSVersion (or LSVersion). All implementations shall ignore unknown TLV blocks. Once the first TLV extension has been defined for a particular subtype, all extensions to that subtype in the future shall require TLV encoding, including any permanent additions to future versions.

The SSVersion or LSVersion field shall be incremented when a permanent extension is defined for all future versions of an LCP subtype, or when a formerly reserved field in the permanent portion of a subtype is defined to have a use within the protocol. The version field should not be incremented for optional extensions.

### 10.10.3 Header size restrictions, and LCP padding

All encapsulating G.9954 LCP headers, short or long format, shall have lengths that are multiples of 4 octets (32 bits). The reserved tag value 0x00 shall be used as padding within the TLV portion of an LCP header to ensure the required alignment of fields (see next paragraph) and to ensure that the total length of the LCP header is a multiple of 4 octets. This requirement minimizes the cost of frame handling by higher layer protocols when headers are removed.

It is further required that all senders shall ensure natural alignment of 16-bit and 32-bit values as measured from the start of the SSType or LSType field. One, two or three octets of padding (value is 0) shall be used each time padding is required to align a following field.

### 10.10.4 Required support

#### 10.10.4.1 Support for optional LCP extensions

Stations supporting G.9954 shall use the Supported Subtypes CSA extension to announce support for optional LCP subtypes, including embedding header subtypes, new control header subtypes and encapsulating subtypes other than LARQ.

For all received subtypes, stations shall ignore unknown extensions, when present, and shall process all known extensions normally as if any unknown extensions were not present.

#### 10.10.4.2 Use of encapsulating headers

Stations shall be capable of removing an unknown encapsulating header and processing the remaining frame as if the unknown header were not present. However, stations shall not add any encapsulating header except for the standard 8-octet LARQ header unless all recipients of the frame are known to support frame lengths long enough to accommodate the extended message size that results when the encapsulating header is present. In addition, encapsulating headers other than the LARQ header should only be sent if all active listeners of the DA of the frame are known to support that type. Active Listener is defined in the G.9954 Link Layer Protocol specification.

A station is considered to support G.9954 or higher only if a CSA message indicating that status has been received from the station within the last two minutes. Stations that go to sleep typically do not generate CSA messages, and will therefore drop from the ranks of "known to be G.9954 or higher", requiring further traffic to be sent with G.9954 default capability limitations assumed for the receiving nodes (for example, the default MTU size) in order to ensure reasonable wakeup behaviour.

This means that encapsulating headers other than 8-octet LARQ headers shall not be used for broadcast or multicast traffic unless CSA messages from every active listener for the broadcast or multicast group indicates support for MTU lengths sufficiently long to accommodate the extended message size that results when the encapsulating header is present. A "station on the wire" is a PNT station that sends Link Integrity frames. The source MAC address used in the Link Integrity frames identifies the station. If no CSA message has been recently received (during the last two minutes) with the same source MAC address, then the station is asleep, and must be treated as G.9954 (see 10.6.5).

Stations shall not add an LCP encapsulating header subtype other than an 8-octet LARQ header if any active listener of the frame's support for MTU lengths is not sufficiently long to accommodate the extended message size that results when the encapsulating header is present. When all active listeners advertise sufficient MTU sizes, a station should not add an LCP encapsulating header subtype other than LARQ unless at least one active listener is known to support the subtype via the Supported Subtypes CSA extension.

Stations should not send an LCP control frame to stations not known to support the subtype. If the MAC destination address is a multicast/broadcast group address, then at least one active listener should be known to understand the subtype.

#### 10.10.4.3 Use of embedding headers

Stations should not send an LCP frame with an embedded payload unless all active listeners are known to understand the subtype (via receipt from all receivers of the new Supported Subtypes extension to CSA control frames with the embedding subtype).

### 10.10.5 TLV extension formats

**Table 10-30/G.9954 – Short-format TLV extension**

| Field | Length | Meaning |
|---|---|---|
| SETag | 1 octet | 1-127. Tag value assigned for extension |
| SELength | 1 octet | Total length of TLV extension excluding the tag and length octets<br>Minimum is 0; maximum is 255 |
| SEData | 0-255 octets[a)] | Additional data for extension |
| [a)] Limited by available space in physical or link layer frame format. | | |

SELength shall not be used as an indicator of the version of information present in the SEData portion of the TLV.

**Table 10-31/G.9954 – Long-format TLV extension**

| Field | Length | Meaning |
|---|---|---|
| LETag | 2 octets | 32768-65535. Tag value assigned for extension |
| LELength | 2 octets | Total length of TLV extension excluding the tag and length octets<br>Minimum is 0; maximum is 65526 |
| LEData | 0-65526 octets[a)] | Additional data for extension |
| [a)] Limited by available space in physical or link layer frame format. | | |

LELength shall not be used as an indicator of the version of information present in the LEData portion of the TLV.

**Table 10-32/G.9954 – Pad, may be used with all TLV extensions**

| Field | Length | Meaning |
|---|---|---|
| LCP_Ext_Pad | 1 octet | = 0 (LCP_EXT_PAD). May be repeated up to three times in succession |

**Table 10-33/G.9954 – Example: Short-format frame with TLV extension**

| Field | Length | Meaning |
|---|---|---|
| DA | 6 octets | Destination Address |
| SA | 6 octets | Source Address |
| Ethertype | 2 octets | 0x886c |
| SSType | 1 octet | = x |
| SSLength | 1 octet | Number of additional octets in the control header, starting with the SSVersion field and ending with the second (last) octet of the Next Ethertype field. The total length from SSType through Next Ethertype must be a multiple of 2 (natural alignment of Next Ethertype) with SSLength being an even integer, or a multiple of 4 (encapsulating header), with SSLength mod 4 equal to 2. |
| SSVersion | 1 octet | = x |
| Fixed/known data for SSVersion | | |
| SETag | 1 octet | Tag value assigned for extension |
| SELength | 1 octet | Total length of TLV extension excluding the tag and length octets<br>Minimum is 0; maximum is 255 |
| SEData | 0-255[a] octets | Additional data for extension |
| [Additional TLV extensions] | | |
| [padding if necded] | 0-3 octets | Must be zero |
| Next Ethertype | 2 octets | |

[a] Limited by available space in physical or link layer frame format.

### 10.10.6 CSA extensions

#### 10.10.6.1 CSA extension for supported optional subtypes

The following extension (in Table 10-34) is defined for CSA frames to allow implementations to advertise support for each optional subtype. Optional subtypes are defined as those subtypes that are defined, but not required, by some version of the PNT specification. Initially, this will include any new G.9954 subtypes for which support is not required in G.9954 devices. Rather than attempt to conserve a little space, all frame types are treated as 16-bit integers, most significant octet sent first.

**Table 10-34/G.9954 – Supported subtypes TLV extension for CSA**

| Field | Length | Meaning |
|---|---|---|
| SETag | 1 octet | = CSA_SUBTYPES_TAG |
| SELength | 1 octet | Total length of TLV extension excluding the tag and length octets<br>2 × number of advertised subtypes |
| Subtype1 | 2 octets | First supported optional subtype as a 16-bit integer (may be short or long subtype) |
| [Subtype2,…,n] | 2 × (n – 1) octets | Additional optional subtypes supported by the implementation. |

### 10.10.6.2 CSA extension for subtype parameters

The following extension (in Table 10-35) is defined for CSA frames to allow implementations to advertise implementation-specific parameters for individual LCP subtypes. Not all LCP frame types will require additional parameters. The definition of parameters is subtype-dependent, and outside the scope of this Recommendation.

**Table 10-35/G.9954 – Subtype parameters TLV extension for CSA**

| Field | Length | Meaning |
|---|---|---|
| SETag | 1 octet | = CSA_PARAMS_TAG |
| SELength | 1 octet | Total length of TLV extension excluding the tag and length octets<br>Minimum is 3; maximum is 255 |
| Subtype | 2 octets | The subtype for which additional parameters are being specified |
| Parameter Data | 1+ octets | Implementation-specific data |

### 10.10.6.3 Vendor-specific extension, Short format

The following extension (in Table 10-36) is defined for all extensible subtypes.

**Table 10-36/G.9954 – Vendor-specific short-format TLV extension**

| Field | Length | Value/Meaning |
|---|---|---|
| SETag | 1 octet | VENDOR_SHORT_TAG |
| SELength | 1 octet | Total length of TLV extension excluding the tag and length octets<br>Minimum is 4; maximum is 255 |
| SVsOUI | 3 octets | An IEEE-assigned Organizationally Unique Identifier |
| SVsData | 0-251 octets[a)] | Vendor-specific data for extension |
| [a)] Limited by available space in physical or link layer frame format. | | |

### 10.10.6.4 Vendor-specific extension, Long format

The following extension (in Table 10-37) is defined for all extensible subtypes.

**Table 10-37/G.9954 – Vendor specific long format TLV extension**

| Field | Length | Value/Meaning |
|---|---|---|
| LETag | 2 octets | VENDOR_LONG_TAG |
| LELength | 2 octets | Total length of TLV extension excluding the tag and length octets<br>Minimum is 4; maximum is 65526 |
| LVsOUI | 3 octets | An IEEE-assigned Organizationally Unique Identifier |
| LVsData | 0-65522 octets[a)] | Vendor-specific data for extension |
| [a)] Limited by available space in physical or link layer frame format. | | |

### 10.10.7 Subtype and tag assignments

Tables 10-38 and 10-39 list the current (and planned) assignment of LCP subtypes, and Tag values for LCP extensions.

**Table 10-38/G.9954 – Subtype assignments**

| Subtype name | Value | Use |
|---|---|---|
| Reserved | 0 | Reserved |
| SUBTYPE_RATE | 1 | Rate request protocol |
| SUBTYPE_LINK | 2 | Link integrity protocol |
| SUBTYPE_CSA | 3 | Capabilities and status announcement protocol |
| SUBTYPE_LARQ | 4 | Limited automatic repeat request protocol |
| SUBTYPE_VENDOR_SHORT | 5 | Vendor-specific short-format header |
| SUBTYPE_FRAME_BURSTING | 6 | Frame Bursting Protocol |
| SUBTYPE_master_SELECTION | 7 | Dynamic master selection protocol |
| SUBTYPE_TIMESTAMP_REPORT | 8 | Timestamp Report Indication |
| Reserved | 9-127 | Reserved/Unassigned |
| Reserved | 128-255 | Reserved for long message type |
| Reserved | 32768 | Reserved |
| SUBTYPE_VENDOR_LONG | 32769 | Vendor-specific long-format subtype |
| SUBTYPE_CERT | 32770 | Certification protocol |
| SUBTYPE_RS | 32771 | Reed-Solomon header |
| SUBTYPE_MAP | 32772 | MAP Synchronization Protocol |
| SUBTYPE_REGISTRATION | 32773 | Network admission control (registration) protocol |
| SUBTYPE_FLOW_SIGNALLING | 32774 | Flow Signalling Protocol |
| Reserved | 32775-65535 | Reserved/Unassigned |

**Table 10-39/G.9954 – Tag assignments**

| Tag name | Value | Use |
|---|---|---|
| LCP_EXT_PAD | 0 | Single octet (no length field), padding for alignment, all subtypes |
| VS_SHORT_TAG | 1 | Vendor-specific extension, short format, all subtypes |
| CSA_SUBTYPES_TAG | 2 | List of supported optional subtypes, CSA only |
| CSA_PARAMS_TAG | 3 | Parameters for a subtype, CSA only |
| CSS_TAG | 4 | Collision Signalling Sequence (see 10.12), CSA only |
| RRCF_RS_TAG | 2 | Reed-Solomon extension (see 10.11.7), Rate Negotiation only |
| RRCF_CID_TAG | 3 | Logical channel ID extension (see 10.4.2), Rate Negotiation only |
| FS_PARAMS_TAG | 2 | Flow parameters (see 10.17.1.1), Flow Signalling only |
| FS_CLASSIFIER_TAG | 3 | Flow classification filter (see 10.17.1.2), Flow Signalling only |
| VS_LONG_TAG | 32769 | Vendor-specific extension, long format, all subtypes |

### 10.10.8 Reservation of LCP subtypes and TLV tags for experimental use

Small ranges of short and long format values for LCP subtypes and TLV Extension tags should be reserved for experimental use. The suggest range for short-format values is 124 through 126 (3 values), inclusive. The suggested range for long-format values is 65280 through 65534 (255 values). These ranges apply to both subtypes and tags. These values are reserved exclusively for development purposes, and shall not be including as part of a PNT-compliant implementation.

## 10.11 Reed-Solomon coding with intra-frame interleaving (Optional)

This clause describes the use of an optional Reed-Solomon code and intra-frame interleaving of bytes.

### 10.11.1 Embedded Reed-Solomon codewords

The Reed-Solomon codeword checkbytes are embedded within the PNT packet with a Tag-Length-Value (TLV) encapsulating header; the original payload shall be unchanged and will follow the checkbytes. This maintains backward compatibility with G.9951/2 nodes; PNT nodes that do not perform RS decoding can ignore the encapsulating header and recover the original payload (assuming no transmission errors).

TLV extensions allow Reed-Solomon coding and decoding to be implemented in a device driver, as long as the receiver hardware still sends packets that fail the FCS and CRC-16 checks up to the logical layer in the driver for possible error correction.

If the RS decoding is performed in a device driver above PNT demodulator, the demodulator should (we do mean "should", not "shall") also pass the FCS and CRC-16 values to the RS decoder in order to verify that the correction was successful. If the recalculated FCS and CRC-16 fail after the payload has been corrected via RS, the receiver may wish to flag the packet as uncorrected and ask for retransmission.

### 10.11.2 Reed-Solomon symbol size

The symbol size shall be 8 bits, resulting in a code based on GF(256). This limits the maximum codeword size to 255; a PNT packet may contain several codewords. The primitive polynomial and generator polynomials are identical to those used for ITU-T Rec. G.992.1.

The arithmetic is performed in the Galois Field GF(256), where $\alpha$ is a primitive element that satisfies the primitive binary polynomial $x^8 + x^4 + x^3 + x^2 + 1$. A data byte ($d_7$ , $d_6$, ... , $d_1$, $d_0$) is identified with the Galois Field element $d_7\alpha^7 + d_6\alpha^6 ... + d_1\alpha + d_0$.

### 10.11.3 Generator polynomial

$G(X) = \Pi(X + \alpha^i)$ is the generator polynomial of the Reed-Solomon code, where the index of the product runs from $i = 0$ to $R - 1$. X is a unit byte delay, and $R$ is the number of check bytes per codeword.

### 10.11.4 Number of check bytes per codeword: Range of values

$R$ can be one of 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, or 20. Implementations do not need to encode or decode all of these allowed values of $R$; when stations announce the ability to perform RS encoding and decoding, they shall also announce the set of $R$ values they support for encoding and decoding.

### 10.11.5 Interleaving

Because the length of PNT packets can range from 64 to 1522 and beyond, a packet can contain several codewords. These codewords could be transmitted sequentially inside a single packet, but a better solution is to interleave the codewords within the packet, giving added protection from bursty errors.

Interleaving shall be applied only within a single packet rather than span across multiple packets. The interleaver resets at the beginning of each packet.

The range of interleaving depths $D$ shall be 1, 2, 4, 8, 16, 32, 64; the interleaver depths vary by a factor of 2. An interleaving depth of 64 allows a packet length of slightly over 16 000 bytes, although the specification limits the packet length to $1024 \times N$ octets where $N$ is the number of bits per symbol (for 2-Mbaud modulation).

The interleaving method is a simple write-by-columns, code-by-rows block interleave method. The transmission order of the original payload bytes is not affected; the interleaving is used conceptually for the calculation of the redundant bytes.

*Interleaving example*

An example of interleaving is shown in the next few paragraphs using a packet payload of 15 bytes, with $R = 2$ and $D = 4$.

Below is the original payload containing 15 bytes $S_1$ through $S_{15}$.

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $S_1$ | $S_2$ | $S_3$ | $S_4$ | $S_5$ | $S_6$ | $S_7$ | $S_8$ | $S_9$ | $S_{10}$ | $S_{11}$ | $S_{12}$ | $S_{13}$ | $S_{14}$ | $S_{15}$ |

Below is a representation of the payload as a two-dimensional array; the number of rows is equal to the interleave depth $D$.

| | | | |
|---|---|---|---|
| $S_1$ | $S_5$ | $S_9$ | $S_{13}$ |
| $S_2$ | $S_6$ | $S_{10}$ | $S_{14}$ |
| $S_3$ | $S_7$ | $S_{11}$ | $S_{15}$ |
| $S_4$ | $S_8$ | $S_{12}$ | |

There are now 4 (= $D$) codewords. Each RS codeword now reads across rows; the first codeword consists of bytes $S_1$, $S_5$, $S_9$, and $S_{13}$. Each codeword has at most 4 bytes, which equals `ceil`(15/4) or `ceil`(K/D) where K equals the payload length, and `ceil`($x$) is the minimum integer larger than $x$; the last payload has 3 bytes because $K$ is not an integral number of $D$.

Below the Reed-Solomon check bytes are appended to each codeword. These check bytes are labeled $C_{\text{codeword-index,checkbyte-index}}$.

| | | | | | |
|---|---|---|---|---|---|
| $S_1$ | $S_5$ | $S_9$ | $S_{13}$ | $C_{1,1}$ | $C_{1,2}$ |
| $S_2$ | $S_6$ | $S_{10}$ | $S_{14}$ | $C_{2,1}$ | $C_{2,2}$ |
| $S_3$ | $S_7$ | $S_{11}$ | $S_{15}$ | $C_{3,1}$ | $C_{3,2}$ |
| $S_4$ | $S_8$ | $S_{12}$ | $C_{4,1}$ | $C_{4,2}$ | |

After the check bytes are calculated, the packet can be transmitted. As described earlier, the check bytes shall be transmitted separately in an encapsulating header which will be described in detail later. Notice that the payload is transmitted in its original byte ordering.

Payload transmission order for this example:

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $S_1$ | $S_2$ | $S_3$ | $S_4$ | $S_5$ | $S_6$ | $S_7$ | $S_8$ | $S_9$ | $S_{10}$ | $S_{11}$ | $S_{12}$ | $S_{13}$ | $S_{14}$ | $S_{15}$ |

Check-byte transmission order for this example is:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $C_{1,1}$ | $C_{2,1}$ | $C_{3,1}$ | $C_{4,1}$ | $C_{1,2}$ | $C_{2,2}$ | $C_{3,2}$ | $C_{4,2}$ |

Check-byte transmission order:

In general, check bytes shall be transmitted in the following order: $C_{i,j}$ where $i$ is the codeword index, $j$ is the check-byte index, and the $i$ (codeword) index varies quickest. If the number of check bytes (= $R \times D$) is not a multiple of 4, then two zero bytes shall be appended to the check bytes so that the payload, or next TLV extension, starts on a 4-byte boundary.

### 10.11.6 Indicating the Redundancy Parameters *R* and *D*

Packet lengths can vary on a packet by packet basis, and the value of $D$ may also have to vary in order to ensure that any single codeword does not exceed the 255-symbol limit. Furthermore, it is desirable to give implementors a wide range of flexibility in determining the amount of redundancy to provide. The above two considerations motivate a mechanism to allow a transmitter to vary $R$

and *D* on a packet-by-packet basis; in general a transmitter may vary *R* and *D*, with *D* chosen to limit the codeword length, and *R* chosen to provide the desired redundancy.

The mechanism must transmit the *R* and *D* parameters in a robust manner, as any error in *R* or *D* will render the entire packet uncorrectable. The need for robustness is complicated by the fact that the *R* and *D* parameters will often be transmitted at a payload rate which is higher due to the increased SNR of RS coding; therefore the *R* and *D* parameters themselves to be redundantly transmitted. To avoid forcing a minimum RS decoding capability in all transceivers, these parameters, along with the Tag, Length, and Payload Length Values, are simply repeated 3 times; receivers can vote on the three received sets.

*Format of Reed-Solomon protocol header*

The length of the encapsulating header must be a multiple of 4 octets, measured from the SSType field through the Next Ethertype field inclusively. The header consists of 3 copies of the SSType, SSLength, SSVersion, and SSParams, followed in turn by a set of check bytes, followed by a Next Ethertype field. The set of check bytes shall be zero-padded, if necessary, to ensure that the length of the header will be a multiple of 4 bytes.

SSVersion has two fields. One field shall be the version of RS encoder being used (0 at the time of this edition of G.9954) and another field shall be the length of the packet modulo 16. The length encoded here is the sum of all bytes starting at the Reed-Solomon SSType and ending at the end of the payload that Reed Solomon encoding covers, which would be the entire G.9954 packet excluding the FCS and CRC-16.

*RSParams*

Table 10-40 shows the format of the RSParams octet.

**Table 10-40/G.9954 – RSParams octet format**

| Bit 7 (MSB) | Bit 4 | Bit 3 | Bit 0 (LSB) |
|---|---|---|---|
| R field | | D field | |

Table 10-41 shows the encoding of the R field.

**Table 10-41/G.9954 – R field encoding**

| Field value (Bit 7..Bit 4) | R |
|---|---|
| 0000 | 0 |
| 0001 | 2 |
| 0010 | 4 |
| 0011 | 6 |
| 0100 | 8 |
| 0101 | 10 |
| 0110 | 12 |
| 0111 | 14 |
| 1000 | 16 |
| 1001 | 18 |
| 1010 | 20 |

Table 10-42 shows the encoding of the D field.

**Table 10-42/G.9954 – D field encoding**

| Field value (Bit 3..Bit 0) | D |
|---|---|
| 0000 | 1 |
| 0001 | 2 |
| 0010 | 4 |
| 0011 | 8 |
| 0100 | 16 |
| 0101 | 32 |
| 0110 | 64 |

**Table 10-43/G.9954 – Format of TLV header, long form**

| Field | Length | Meaning |
|---|---|---|
| DA | 6 octets | Destination Address (from original Ethernet PDU) |
| SA | 6 octets | Source Address (from original Ethernet PDU) |
| Ethertype | 2 octets | 0x886c |
| LSType | 2 octets | SUBTYPE_RS = 32771. Reed-Solomon encapsulating header type (provisional) |
| LSLength | 2 octets | Number of additional octets in the RS header, starting with the SSVersion field and ending with the second (last) octet of the Next Ethertype field |
| RSVersion | 1 octet | = 0-15, overloading the version to encode the payload length modulo 16 |
| RSParams | 1 octet | RS Redundancy Parameters (4-bits each for D, R as already proposed) This field has 2 replicated bytes. |
| LSType2 | 2 octets | Duplicate of LSType |
| LSLength2 | 2 octets | Duplicate of LSLength |
| RSVersion2 | 1 octet | Duplicate of RSVersion |
| RSParams2 | 1 octet | Duplicate of RSParams |
| LSType3 | 2 octets | Duplicate of LSType |
| LSLength3 | 2 octets | Duplicate of LSLength |
| SSVersion3 | 1 octet | Duplicate of RSVersion |
| RSParams3 | 1 octet | Duplicate of RSParams |
| RSCheckBytes | D*R octets (possibly padded to a multiple of 4) | The array of computed check bytes.<br>Order of transmission: ($C_{1.1}$, $C_{2.1}$ .. $C_{D.1}$, $C_{1.2}$, $C_{2.2}$ .. $C_{D.2}$, … $C_{1.R}$ ..$C_{D.R}$)<br>Two additional zero bytes may follow, to pad to multiple of 4 bytes.<br>The RS payload coding starts with the next field, typically "Next_Ethertype". |
| Next Ethertype | 2 octets | From "original" Ethernet PDU (could be 886c, with a LARQ header) |
| Payload | Min.TBD octets | From original Ethernet PDU payload |
| FCS | 4 octets | Frame Check Sequence |
| CRC-16 | 2 octets | PNT Frame Check Sequence |

### 10.11.7 Receiver indication of desired encoding. TLV Extension to the LCP SUBTYPE_RATE Subtype

It is the receiver that monitors the Packet Error Ratio, and is therefore the best qualified to guide the selection of Reed-Solomon redundancy. A mechanism to have a receiver indicate the desired redundancy to a remote transmitter is described below. It is a TLV extension of the current Rate Request Control Frame.

Three additional parameters are included for each band: One is an enhanced payload rate (Bandn_EPR) and the other two parameters indicate a minimum redundancy that will allow transmission at that enhanced payload rate. The Band*n*_EPR format is the same as that for non-RS encoded Band*n*_PE.

The minimum redundancy is specified by two octets. The first octet specifies a desired number of redundant bytes per RS codeword; the remote transmitter should encode all payloads with this number of redundant bytes. As the number of redundant bytes is a multiple of 2, this field shall be encoded as R/2.

The second octet that specifies the desired redundancy is a maximum payload size per codeword. The remote transmitter should restrict the payload to never exceed this length, by increasing D if necessary.

Notice that SSVersion is 1, to indicate that this has a different format from the current RRCF format (see Table 10-44). This shall only be sent if capabilities exchange will show that RS coding is supported.

**Table 10-44/G.9954 – Rate request control frame definition with Reed-Solomon extension**

| Field | Length | Meaning |
|---|---|---|
| DA | 6 octets | Destination Address |
| SA | 6 octets | Source Address |
| Ethertype | 2 octets | 0x886c |
| SSType | 1 octet | = 1 |
| SSLength | 1 octet | Number of additional octets in the control header, starting with the SSVersion field and ending with the second (last) octet of the Next Ethertype field. The minimum value of SSLength is 8 for SSVersion 0. |
| SSVersion | 1 octet | = 0 [1?] |
| OpCode | 1 octet | Operation code for this control message. See Table 10-6 for definitions. |
| NumBands | 1 octet | Number of bands specified in this control. […]. |
| NumAddr | 1 octet | Number of addresses specified in the payload of this control message. NumAddr may be zero. […]. |
| Band1_PE | 1 octet | 2-MBaud, 7-MHz carrier: The PE value that should be used to send data when the 2-MBaud band is selected. […] |
| Band1_rank | 1 octet | The rank order of the ReqDAs' preference for this band […] |
| Band2_PE | 1 octet | Optional: Only present if NumBands ≥ 2. […] |
| Band2_rank | 1 octet | Optional: Only present if NumBands ≥ 2. […] |
| RefAddr1 | 6 octets | Optional: Present if NumAddr ≥ 1. […] |
| RefAddr2 | 6 octets | Optional: Present if NumAddr ≥ 2. […] |

**Table 10-44/G.9954 – Rate request control frame definition with Reed-Solomon extension**

| Field | Length | Meaning |
|---|---|---|
| • • • | | [additional instances of RefAddr, until the number of RefAddr fields equals NumAddr] |
| SETag | 1 octet | = 2. Optional RS values for rate negotiation |
| SELength | 1 octet | Total length of option excluding the tag and length octets, and pad.<br>Must be 2 + 4 × Numbands; Minimum is 6 |
| Band1_EPR | 1 octet | Enhanced payload rate to use when Reed-Solomon coding is used at the target redundancy specified in the next field |
| Band1_RSR | 1 octet | Number of redundant bytes per codeword when Reed-Solomon coding is used |
| Band1_Kmax | 1 octet | Maximum payload size per codeword |
| Band1_Pad [suggest Band1_Rdesired] | 1 octet | For alignment and possible extensions. = 0 if unused.<br>[Desired upper limit for total number of total redundant bytes, allowing reduced redundancy per code-word for longer frames.] |
| • • • | | [additional instances of RS coding params, if Numbands ≥ 2] |
| Pad | 2 octets | Pad to make encapsulating header a multiple of 4 octets |
| Next Ethertype | 2 octets | = 0 |
| Pad | | To reach minFrameSize if required |
| FCS | 4 octets | Frame Check Sequence |
| CRC-16 | 2 octets | PNT Frame Check Sequence |

### 10.11.8 Capabilities Announcement

The ability of a station to encode and decode packets shall be transmitted in the CSA_SUBTYPES tag extension to the CSA frame; via member fields. An example of this extension is in Table 10-45.

**Table 10-45/G.9954 – Example TLV extension for CSA, announcing RS capability**

| Field | Length | Meaning |
|---|---|---|
| SETag | 1 octet | = CSA_SUBTYPES_TAG. |
| SELength | 1 octet | Total length of TLV extension excluding the tag and length octets.<br>2 × number of advertised subtypes (n). |
| Subtype | 2 octets | SUBTYPE_RS_LONG (32771) |
| Additional Subtypes | 2 × (n – 2) octets | Additional optional subtypes supported by the implementation |

In addition to announcing support for the Reed-Solomon subtype, the explicit capabilities of a station announcing RS capability are sent in a CSA Parameters extension. This indication in a CSA extension for subtype parameters. The format of this extension is in Table 10-46.

**Table 10-46/G.9954 – RS subtype parameters TLV extension for CSA**

| Field | Length | Meaning |
|---|---|---|
| SETag | 1 octet | = CSA_PARAMS_TAG |
| SELength | 1 octet | = 6 |
| Subtype | 2 octets | SUBTYPE_RS (32771) |
| Supported Encoding R Value Bit Mask | 2 octets | *First octet*<br>Bit R<br>0 2<br>1 4<br>2 6<br>3 8<br>4 10<br>5 12<br>6 14<br>7 16<br>*Second octet*<br>Bit R<br>0 18<br>1 20<br>2 through 7, reserved |
| Supported Decoding R Value Bit Mask | 2 octets | *First octet*<br>Bit R<br>0 2<br>1 4<br>2 6<br>3 8<br>4 10<br>5 12<br>6 14<br>7 16<br>*Second octet*<br>Bit R<br>0 18<br>1 20<br>2 through 7, reserved |
| Parameter Data | 1+ octets | Implementation-specific data |

## 10.12 Collision Management Protocol

The Collision Management Protocol defines a mechanism to dynamically assign unique, fixed sequences of collision signal slot values to stations in order to control access latency for devices using PHY priority 7 for very low latency traffic (such as voice).

### 10.12.1 Terms and acronyms

**10.12.1.1 active CSS client**: A CSS Client that has a CSS assignment and is using CSS for frame transmission.

**10.12.1.2 channel**: A logical flow, the transmitted series of frames from a single instance of an application, such as frames containing one simplex stream of digitized voice.

**10.12.1.3 Collision Signalling Sequence (CSS)**: A set of DFPQ signal slot assignments, the management of which can provide latency bounds for highest priority frames.

**10.12.1.4 CSS client**: Any station that participates in the assignment of CSS sequences using the CSS TLV Extension. A CSS Client chooses its own CSS.

**10.12.1.5 CSS extension**: A TLV structure added to CSA Messages containing information to support distributed assignment of CSS values to stations.

**10.12.1.6 CSS protocol**: A protocol for distributing CSS values to stations via a CSA extension.

**10.12.1.7 multichannel client**: A CSS Client sending multiple independent streams of frames at PHY priority 7. For example, a gateway device supporting multiple independent (non-aggregated) streams. Some CSS values may provide better service for multichannel clients.

**10.12.1.8 single-channel client**: A CSS Client that transmits a single stream of frames at PHY priority 7. A one-line PNT telephone is an example of a typical single channel device.

### 10.12.2 Collision Signalling Sequence

A Collision Signalling Sequence (CSS) is an ordered set of two-bit values [s1, s2, …sN] used to control the behaviour of an PNT DFPQ MAC following collisions. A value in the range [0,2] for s<x> indicates a specific signalling slot to be used following the <x>th collision, while the value of 3 indicates the use of a random value chosen by the station at the time of the collision. A signal slot value in a sequence shall be used once per frame at most. If a station encounters more collisions than values listed in the sequence, then selection reverts to a random slot selection until the frame is either transmitted or dropped (behaving as if the sequence had a series of trailing 3s).

The number of supported Active CSS Clients is 27, using only the first three collision signal slots, with the fourth and later signal slots specified as 3, indicating random assignment. The unique portion of an assigned CSS value shall cover three collision signal slots, such that a standard CSS value will have s1-s3 in the ranges 0-2 and s4-s8 set to 3.

The set of CSS values are enumerated, and each sequence given an explicit rank based on the order of frame transmissions following collisions that occur between stations having unique CSS assignments. CSS values are assigned so as to minimize the number of collisions. The ordering of the first 27 sequences is shown in Table 10-47. This ordering shall be used by CSS Clients to choose the next CSS to be assigned from the set of unused values. The first three multichannel stations may be assigned the first three sequences in order to minimize repeated collisions for each of the streams.

**Table 10-47/G.9954 – CSS values in order of assignment**

| CSS sequence number (in assignment order) | Sequence s1,s2,s3,s4,s5,s6,s7,s8 | Transmit rank |
|---|---|---|
| 1[a)] | 0,0,0,3,3,3,3,3 | 1 |
| 2[a)] | 1,0,0,3,3,3,3,3 | 10 |
| 3[a)] | 2,0,0,3,3,3,3,3 | 19 |
| 4 | 0,1,0,3,3,3,3,3 | 4 |
| 5 | 0,2,0,3,3,3,3,3 | 7 |
| 6 | 1,1,0,3,3,3,3,3 | 13 |
| 7 | 1,2,0,3,3,3,3,3 | 16 |
| 8 | 2,1,0,3,3,3,3,3 | 22 |
| 9 | 2,2,0,3,3,3,3,3 | 25 |
| 10 | 0,0,1,3,3,3,3,3 | 2 |
| 11 | 0,0,2,3,3,3,3,3 | 3 |
| 12 | 0,1,1,3,3,3,3,3 | 5 |
| 13 | 0,1,2,3,3,3,3,3 | 6 |
| 14 | 0,2,1,3,3,3,3,3 | 8 |
| 15 | 0,2,2,3,3,3,3,3 | 9 |
| 16 | 1,0,1,3,3,3,3,3 | 11 |
| 17 | 1,0,2,3,3,3,3,3 | 12 |
| 18 | 1,1,1,3,3,3,3,3 | 14 |
| 19 | 1,1,2,3,3,3,3,3 | 15 |
| 20 | 1,2,1,3,3,3,3,3 | 17 |
| 21 | 1,2,2,3,3,3,3,3 | 18 |
| 22 | 2,0,1,3,3,3,3,3 | 20 |
| 23 | 2,0,2,3,3,3,3,3 | 21 |
| 24 | 2,1,1,3,3,3,3,3 | 23 |
| 25 | 2,1,2,3,3,3,3,3 | 24 |
| 26 | 2,2,1,3,3,3,3,3 | 26 |
| 27 | 2,2,2,3,3,3,3,3 | 27 |
| [a)] Multichannel stations should use these values, if more than one is operational. | | |

### 10.12.3 CSA extension to support CSS assignment

The CSS assignment protocol uses the CSA protocol with a TLV-based extension. All stations requiring CSS assignment shall implement the CSA protocol and shall support the CSA priority mapping functions.

#### 10.12.3.1 CSS flag assignments

Each CSS value is assigned a bit flag in a field of contiguous bits, with the first CSS value assigned the least significant bit when the field is treated as an unsigned integer in network byte order (see Table 10-48).

**Table 10-48/G.9954 – CSS flag set, showing assignment of bits to CSS sequence values**

| Octet | Field | Length [bits] | Description |
|---|---|---|---|
| CSSFlags0 | Reserved | 5 | For CSS Extension version 0, sent as 0, ignored when received. |
| | CSS_Seq27 | 1 | Station is (was) using CSS Sequence number 27. |
| | CSS_Seq26 | 1 | Station is (was) using CSS Sequence number 26. |
| | CSS_Seq25 | 1 | Station is (was) using CSS Sequence number 25. |
| CSSFlags1 | CSS_Seq24 | 1 | Station is (was) using CSS Sequence number 24. |
| | CSS_Seq23 | 1 | Station is (was) using CSS Sequence number 23. |
| | CSS_Seq22 | 1 | Station is (was) using CSS Sequence number 22. |
| | CSS_Seq21 | 1 | Station is (was) using CSS Sequence number 21. |
| | CSS_Seq20 | 1 | Station is (was) using CSS Sequence number 20. |
| | CSS_Seq19 | 1 | Station is (was) using CSS Sequence number 19. |
| | CSS_Seq18 | 1 | Station is (was) using CSS Sequence number 18. |
| | CSS_Seq17 | 1 | Station is (was) using CSS Sequence number 17. |
| CSSFlags2 | CSS_Seq16 | 1 | Station is (was) using CSS Sequence number 16. |
| | CSS_Seq15 | 1 | Station is (was) using CSS Sequence number 15. |
| | CSS_Seq14 | 1 | Station is (was) using CSS Sequence number 14. |
| | CSS_Seq13 | 1 | Station is (was) using CSS Sequence number 13. |
| | CSS_Seq12 | 1 | Station is (was) using CSS Sequence number 12. |
| | CSS_Seq11 | 1 | Station is (was) using CSS Sequence number 11. |
| | CSS_Seq10 | 1 | Station is (was) using CSS Sequence number 10. |
| | CSS_Seq9 | 1 | Station is (was) using CSS Sequence number 9. |
| CSSFlags3 | CSS_Seq8 | 1 | Station is (was) using CSS Sequence number 8. |
| | CSS_Seq7 | 1 | Station is (was) using CSS Sequence number 7. |
| | CSS_Seq6 | 1 | Station is (was) using CSS Sequence number 6. |
| | CSS_Seq5 | 1 | Station is (was) using CSS Sequence number 5. |
| | CSS_Seq4 | 1 | Station is (was) using CSS Sequence number 4. |
| | CSS_Seq3 | 1 | Station is (was) using CSS Sequence number 3. |
| | CSS_Seq2 | 1 | Station is (was) using CSS Sequence number 2. |
| | CSS_Seq1 | 1 | Station is (was) using CSS Sequence number 1. |

### 10.12.3.2 Collision signalling sequence extension for CSA

All Collision Signalling Sequence protocol exchanges use CSA messages containing the CSS Extension.

A Tag-Length-Value (TLV) extension called "CSS Extension" is defined for the CSA protocol, to be inserted following the fixed fields defined for PNT. The CSS Extension is used to announce CSS values among stations.

The CSS Extension shall be added to the CSA message between the last fixed field of the CSA frame (CSA_CurrentRxSet), and the Next Ethertype field of the CSA frame. There is no ordering requirement for the CSS Extension versus other CSA Extensions, but the CSS Extension shall be placed at an offset from the start of the CSA message that has the same alignment, modulo 4, that it

would have had if it were the first extension to follow CSA_CurrentRxSet. The alignment requirement stems from the 32-bit unsigned integer fields in its structure.

Table 10-49 shows the format of the CSS Extension for CSA.

**Table 10-49/G.9954 – CSS Extension for CSA**

| Field | Length | Meaning |
|---|---|---|
| SETag | 1 octet | CSS_TAG indicates a CSS Extension. |
| SELength | 1 octet | 14 = Number of additional octets in this extension. SELength is always 14 for the CSS extension, version 0. |
| CSS_Version | 1 octet | 0 |
| CSS_NumChannels | 1 octet | Number of active Tx channels for the device |
| CSS_CurrentTxSet | 4 octets | Indicates current CSS value used by this station, if any. Set to all zeros if no CSS value is being used by this station. Not aged in the usual CSA sense, except that its value is copied into CSS_OldestTxSet at the end of each CSA period. Flag values are specified in 10.12.3.1. |
| CSS_OldestTxSet | 4 octets | A copy of CSS_CurrentTxSet, made either at the beginning of the current CSA period if no change has taken place or when the value of CSS_CurrentRxSet changed (saves the previous value). Flag values are specified in 10.12.3.1. |
| CSS_CurrentRxSet | 4 octets | The union of CSS flags received from other stations during the current CSA period (CSS_NewRxSet) and flags received during the previous period (CSS_PrevRxSet). Flag values are specified in 10.12.3.1. |

### 10.12.4 CSS assignment procedures

The CSS assignment protocol is a straightforward addition to the basic CSA protocol. Stations implementing CSS assignment shall include the CSS TLV in their transmitted CSA messages unless both CSS_CurrentTxSet and CSS_OldestTxSet have the value of all zeros. Changes in the value of CSS_CurrentTxSet are considered to be changes in advertised CSA state information, and therefore the first CSA message sent with a changed value for CSS_CurrentTxSet shall be retransmitted, per standard CSA protocol operation.

Stations maintain a set of CSS flags using logic similar to that used for the standard CSA flags. CSS_CurrentInUse is defined as the union of CSS_CurrentTxSet and CSS_CurrentRxSet, and indicates the set of CSS values currently in use on the network.

CSS_CurrentTxSet contains the flag for the CSS value currently used by the sending station, or 0 if none is in use.

CSS_OldestTxSet contains either a copy of CSS_CurrentTxSet from the beginning of the current CSA period (i.e., a copy is made once per minute), or the previous value of CSS_CurrentTxSet if the station relinquishes its CSS value, or reassigns itself a new value.

CSS_CurrentRxSet is the union of CSS flags received from other stations during the current CSA period (CSS_NewRxSet) and flags received during the previous period (CSS_PrevRxSet).

#### 10.12.4.1 Use of CSS Extension

Stations implementing the CSS assignment protocol normally include the CSS Extension in each of their outgoing CSA messages. The CSS Extension may be excluded from transmitted CSA messages if both CSS_CurrentTxSet and CSS_OldestTxSet would be zero in the outgoing message.

However, the CSS state information shall continue to be maintained based on received CSS Extensions (or the lack thereof), whether or not the CSS Extension is being transmitted.

#### 10.12.4.2 Assignment of a new CSS value

A station not currently assigned a CSS value chooses the lowest numbered (i.e., lowest assignment order number) CSS value that does not have its flag set, and sets that flag in its outgoing messages. Since this is a change of advertised state information for the station, the first CSA message with a new assignment shall be retransmitted once as per CSA transmission rules.

#### 10.12.4.3 Relinquishing an in-use CSS value

When a station gives up a CSS value, it shall send a new CSA message (retransmitted once), with a CSS Extension that has CSS_OldestTxSet set to the previous value of CSS_CurrentTxSet (indicating the value given up), and CSS_CurrentTxSet set to its new value, either zero if the station no longer has a CSS value, or containing the flag for a newly selected value in the case of a reassignment. Receivers of this information can tell that the previous owner of the CSS value is freeing that value.

#### 10.12.4.4 CSS assignment conflicts

It is possible that two (or more) stations might choose the same CSS value, either at nearly the same time due to coincidence or due to implementation of 10.12.4.6 below, so that multiple stations end up advertising the same flag. If a station receives a CSS extension with a flag in CSS_CurrentTxSet indicating a duplicate assignment, and the number of channels in the received extension (CSS_NumChannels) is greater than or equal to the number of channels advertised by the receiver, then the receiving station shall reassign a new CSS value to itself. If the number of channels in the received CSS extension is greater than the number of channels advertised by the receiver, then the receiver shall choose the next available CSS value. Otherwise, it shall choose the new value randomly from the set of unused CSS values.

#### 10.12.4.5 Use of best CSS values: The step-down rule

When a CSS value is relinquished, other stations will, within at most two minutes, notice that it is free. When the station with the next higher-numbered CSS value notices that the preceding CSS value is free, it shall reassign itself to the new, lower-numbered CSS value, following a short, random, delay interval of not more than two seconds. The purpose of the delay is to prevent misunderstandings concerning the set of in-use CSS values that may arise during the resolution of a conflict between two stations.

As an optimization, if the new CSS value is one of a contiguously numbered block of free values, the station may choose the lowest-numbered CSS value for reassignment, rather than performing a series of sequential reassignments. This situation normally occurs only when a station makes a random choice of a free CSS value following a conflict.

#### 10.12.4.6 Optimization for multichannel clients

A multichannel station may choose one of the first three CSS values even if all are in use, forcing the existing owner to reassign itself a new CSS value, but only if the existing owner has fewer channels in use.

### 10.13 Frame bursting protocol

The frame bursting protocol is required. The purpose of the protocol is to reduce the overhead associated with the physical layer framing format by concatenating frames that share the same DA/SA value with equal or greater priority.

The frame format is in Table 10-50:

**Table 10-50/G.9954 – Frame burst format**

| Field | Length | Meaning |
|---|---|---|
| DA | 6 octets | Destination Address |
| SA | 6 octets | Source Address |
| Ethertype | 2 octets | 0x886c (PNT Link Control Frame) |
| SSType | 1 octet | = SUBTYPE_FRAMEBURST (6) |
| SSLength | 1 octet | 6 |
| SSVersion | 1 octet | = 0 |
| FLH Pad | 1 octet | Reserved; must be sent as 0 and ignored by receiver |
| Packet Length | 2 octets | Length in octets of the first packet, from the first octet following the Packet Length field through the last octet of the data preceding the FCS |
| Next Ethertype | 2 octets | Pre-encapsulation ethertype the first packet being bursted |
| Data#1 | variable | Pre-encapsulation payload data from the frame #1 being bursted |
| FCS#1 | 4 octets | Frame Check Sequence |
| CRC-16#1 | 2 octets | Additional Frame Check Sequence (includes the LLC header) |
| Control Info#2 | 4 or 24 octets | Control Information for second packet according to Table 10-51 or Table 10-52 |
| Next Ethertype#2 | 2 octets | Pre-encapsulation ethertype of second packet being bursted |
| Data#2 | variable | Pre-encapsulation payload data of second packet being bursted |
| FCS#2 | 4 octets | Frame Check Sequence |
| CRC-16#2 | 2 octets | Additional Frame Check Sequence from end of previous CRC-16 |
| • • • | | More bursted packets |
| Control Info#N | 4-24 octets | Control Information for Nth packet according to Table 10-51 or Table 10-52 |
| Next Ethertype#N | 2 octets | Pre-encapsulation ethertype from the frame #N being bursted |
| Data#N | Variable | Pre-encapsulation payload data from the frame #N being bursted |
| FCS#N | 4 octets | Frame Check Sequence |
| CRC-16#N | 2 octets | Additional Frame Check Sequence |
| Burst Termination Trailer | 4 octets | 0xFFFF. Burst termination trailer, indicates the end of the burst |
| Pad | | To reach minFrameSize if required |

**Table 10-51/G.9954 – Short control information**

| Field | Length | Meaning |
|---|---|---|
| FT | 1 octet | The FT of the original packet being bursted |
| SMAC | 1 bit | Synchronous MAC indicator |
| FS | 3 bits | Frame SubType of the original packet being bursted |
| Short | 1 bit | = 1 |

**Table 10-51/G.9954 – Short control information**

| Field | Length | Meaning |
|---|---|---|
| Rsvd | 3 bits | Reserved, must be sent as 0 and ignored by receiver |
| Priority/Flow ID | 4 bits | The priority/flow ID of the original packet being bursted |
| Spare | 4 bits | Spare bits reserved for future use. Must be sent as 0 and ignored by the receiver |
| Packet Length | 2 octets | The original packet length being bursted |

Diagonal shading is used in Table 10-52 to show the decomposition of the field (FT) into bit-fields.

**Table 10-52/G.9954 – Long control information**

| Field | Length | Meaning |
|---|---|---|
| FT | 8 bits | The FT of the original packet being bursted. Encoding of the FT is as defined immediately below: |
| SMAC | 1 bit | Synchronous MAC indicator |
| FS | 3 bits | Frame SubType of the original packet being bursted |
| Short | 1 bit | = 0 |
| Rsvd | 3 bits | Reserved; must be sent as 0 and ignored by receiver |
| Priority/Flow ID | 4 bits | The Priority/Flow ID of the original packet being bursted |
| SI | 4 bits | Scrambler Index of the original packet being bursted |
| PE | 8 bits | Payload Encoding of the original packet being bursted |
| HCS | 8 bits | Header Check Sequence of the original packet being bursted |
| DA | 6 octets | Destination Address of original packet being bursted |
| SA | 6 octets | Source Address of original packet being bursted |
| Ethertype | 2 octets | 0x886c (PNT Link Control Frame) |
| SSType | 1 octet | = SUBTYPE_FRAMEBURST (6) |
| SSLength | 1 octet | 6 |
| SSVersion | 1 octet | = 0 |
| FLH Pad | 1 octet | Reserved; must be sent as 0 and ignored by receiver |
| Packet Length | 2 octets | The original packet length being bursted |

The first packet actually has always the long format of the control information.

The maximum length shall not exceed the maximum allowed time on the wire. The maximum size of the bursted frame shall be negotiated in CSA messages as described in 10.10.6. All frames in a burst shall have the same DA/SA value. When a transmitter is constructing a bursted frame, the priorities of each bursted subframe in an unmanaged network must be equal to or greater than the priority of the first frame. If the priority of a frame is less than the priority of the first subframe of a burst frame, it must not be concatenated into a burst frame, it shall start a new physical layer frame. In a managed network there shall be no limit on the bursted flows between the same DA/SA values.

The burst termination trailer shall be used to indicate the end of the burst.

## 10.14 MAC cycle synchronization

MAC cycle synchronization in SMAC mode shall be performed using the master-generated Media Access Plan (MAP). The MAP indicates the beginning of the MAC cycle and contains the Media Access Plan for the following MAC cycle.

All G.9954 stations shall implement the MAC Cycle Synchronization function in order for synchronous MAC behaviour in a master-controlled network.

### 10.14.1 MAP control frame

In Table 10-53, diagonal shading is used to show the decomposition of a field (TXOP) into sub-fields. The decomposition of the sub-field (TXOPID) into bit-fields is shown by the transition from (diagonal) shading to clear (non-shaded) fields.

**Table 10-53/G.9954 – Map control frame**

| Field | Length | Meaning |
|---|---|---|
| DA | 6 octets | Destination Address = 0xFF:FF:FF:FF:FF:FF |
| SA | 6 octets | Source Address of master device |
| Ethertype | 2 octets | 0x886c (PNT Link Control Frame) |
| LSType | 2 octets | = SUBTYPE_MAP (32772) |
| LSLength | 2 octets | Number of additional octets in the control header, starting with the LSVersion field and ending with the second (last) octet of the Next Ethertype field. Minimum LSLength is 22 for LSVersion 0 |
| LSVersion | 1 octet | = 0 |
| LSPad | 1 octet | Ignored on reception |
| MAPHeader | 12 octets | MAP header as described in Table 10-54 |
| TXOP[1] | 4/6 octets | Transmission Opportunity described by the sub-fields immediately below. The length of the TXOP may be 4 or 6 octets depending on the value of the TXOPCtl sub-field below. |
| TXOPCtl | 2 bits | 0 when the TXOP start time is implicitly defined<br>1 when the TXOP start time is explicitly specified (see TXOPStart sub-field below).<br>2-3 Reserved for future use |
| TXOPLength | 14 bits | The length of the TXOP in 1-µs units. |
| TXOPID | 16 bits | TXOP identifier<br>Composed of the sub-fields described immediately below: |
| SrcDeviceID | 6 bits | Device ID of device at source of flow |
| UniqueFlowID | 10 bits | Unique identifier of the flow originating at the device identified by SrcDeviceID. |
| TXOPStart | 16 bits | TXOP start time measured from the start of the MAC cycle in 1-µs units. This field is optional and is defined only if TXOPCtl is not 0 |
| • • • | | Additional TXOPs |
| TXOP[N] | 4 octets | |

**Table 10-53/G.9954 – Map control frame**

| Field | Length | Meaning |
|---|---|---|
| Next Ethertype | 2 octets | = 0 |
| Pad | | Pad to reach minFrameSize if necessary |
| FCS | 4 octets | Frame Check Sequence |
| CRC-16 | 2 octets | PNT Frame Check Sequence |

In Table 10-54, diagonal shading is used to show the decomposition of the Control Field into bit-fields.

**Table 10-54/G.9954 – MAP control header**

| Field name | Field Size [bits] | Description |
|---|---|---|
| Control Field | 32 | Set of control fields used to control the behaviour of endpoint nodes. The encoding of this field is described immediately below: |
| Modified | 1 | Indicates that the TXOP table defined in this MAP is different from the TXOP table defined in the "previous" MAP where "previous" is defined as the MAP sent in the "previous" MAC cycle with *Sequence Number* one less than the "current" *Sequence Number (*accounting for modulo arithmetic).<br>0 MAP is the same as "previous" cycle<br>1 MAP changed since "previous" cycle<br>This flag may be used by an endpoint for local optimization. |
| CycleLatency RepairMethod | 2 | The cycle latency repair method to be used by end-points when the start of the MAC cycle (as indicated by the arrival time of the MAP) is delayed compared to the scheduled arrival time. For further details, see 7.3.3.<br>0 None – Do not use latency repair techniques at start of cycle.<br>1 Adjust clock – Adjust the clock used to time SMAC transmissions at start of cycle by the delay offset.<br>2-3 Reserved for future use |
| Collision Resolution Method | 2 | The collision resolution (CR) method to be used to resolve collisions during TXOPs defined as contention periods. For further details, see 7.3.7.<br>0 DFPQ (G.9951/2-style CR)<br>1 Bounded-DFPQ<br>2 Reserved for future use |
| SMAC_EXIT | 1 | Exit from synchronous MAC mode. The master shall subsequently cease sending MAPs. This flag is used as an indication for G.9954 devices to enter AMAC mode.<br>0 Remain in SMAC mode<br>1 Exit SMAC mode |

**Table 10-54/G.9954 – MAP control header**

| Field name | Field Size [bits] | Description |
|---|---|---|
| AMAC_DETECTED | 1 | Master detected existence of a device operating in AMAC mode. The method used by the master to detect AMAC nodes is implementation dependent.<br>0 Device operating in AMAC mode NOT detected<br>1 Device operating in AMAC mode detected |
| CP Priority Limit | 3 | Highest priority to be used for transmissions in contention period (CTXOP). May be controlled in order to give priority to CF TXs in an environment (e.g., mixed G.9951/2 and G.9954 network) where CF and CP TXs may collide. Defined values are:<br>0..7 Priority levels |
| MAP_IFG_INCR | 6 | Increment added to CS_IFG (29 µs) in order to determine the size of MAP_IFG (Inter-Frame Gap) planned between TXOPs by the master. MAP_IFG is defined by the relation:<br>*MAP_IFG = CS_IFG + MAP_IFG_INCR*<br>MAP_IFG silence shall be guaranteed by each endpoint at the end of its TXOP. MAP_IFG_OFFSET is measured in 500-ns units. |
| Reserved | 16 | Reserved for future use. Shall be sent as 0 and ignored by the receiver |
| Reserved | 32 | Reserved for future use. Shall be sent as 0 and ignored by the receiver |
| SequenceNumber | 16 | MAP sequence number. Modulo counter that is incremented each MAC cycle |
| NumTXOPs | 16 | Number of Entries in allocation map. The minimum number of entries in a MAP is normally 2 (one entry for the MAP and the second entry for the UNALLOCATED TXOP). When the SMAC_EXIT flag is set, the number of entries in the MAP may be zero.<br>The maximum number of entries is limited by the maximum size of the MAP control frame as described above. |

### 10.14.2 Terms and parameters

#### 10.14.2.1 Timers

SYNC_Timer: A free running timer with a period of 150 ms.

This timer is used to detect loss of synchronization with the master-generated MAC cycle. The timer is activated upon entry to SMAC mode and cancelled upon leaving SMAC mode.

### 10.14.3 MAC cycle synchronization protocol

#### 10.14.3.1 Receive MAP control frame

If the G.9954 device is currently in AMAC mode, the periodic SYNC_Timer should be armed and the system state changed to SMAC mode.

If the G.9954 device is already in SMAC mode, the SYNC_Timer should be re-armed to count a new SYNC timeout period.

Control information communicated in the MAP should be used to update system state variables used by the MAC processor.

#### 10.14.3.2 SYNC_Timer timeout

When a SYNC_Timer timeout occurs, this indicates that a MAP was not received for the SYNC_Timer period and that SYNC_LOSS has occurred.

The current MAC mode should be changed to AMAC mode, system state variables updated.

### 10.15 Network Admission Control (Registration) Protocol

In a master-controlled network, a G.9954 device that supports flows with QoS Contracts is required to perform the following procedures in order to enter the network:

- Synchronization – Wait for periodic MAP transmissions from the master.
- Registration – Locate transmission opportunities in the MAP for the transmission of registration protocol messages and perform registration with the master.

The synchronization procedure involves waiting for the reception of a periodic MAP transmission from the currently assigned master. Once a MAP is received, a G.9954 device that wishes to join the network is able to locate available transmission opportunities and proceed with the registration procedure.

The registration procedure consists of a request-response sequence of transactions between the master and the registering device. The registration procedure is used to authenticate a device for network entry, to assign it a unique device identifier and to download network configuration information.

#### 10.15.1 Registration opportunities

Once a device is synchronized with the MAC cycle, the device is required to locate transmission opportunities that will allow it to bootstrap the registration process. Such transmission opportunities are identified in the MAP by either unallocated (spare) bandwidth or explicit REGISTRATION opportunities. For more information on the identity of REGISTRATION TXOPs, see 7.3.3.4.2.

The master guarantees to provide either sufficient spare bandwidth or to allocate REGISTRATION TXOPs at least once every REG_PERIOD. The REGISTRATION transmission opportunity is used to advertise an intention to register. This intention is expressed by sending a REG_REQUEST message to the master.

Devices contend for access to the REGISTRATION transmit opportunities.

#### 10.15.2 Registration and authorization control

Registration is the process performed to allow a G.9954 device to request the bandwidth reservation. Only after a device has registered with the master can it reserve bandwidth through explicit flow setup requests with the master.

The registration procedure involves a request-response sequence, whereby a G.9954 device requests to be registered with the master by sending a REG_REQUEST message containing the device's MAC address as well as other identifying characteristics, such as authentication key and a set of capability parameters. Upon receiving a REG_REQUEST message, the master is responsible for authorizing the entry of the requesting device and, if authorization is successful, for allocating resources to the registered device.

Authorization is performed by checking that the device, identified by its MAC address and possibly other identifying information (e.g., authentication key), is valid and the device is authorized to join the home network controlled by the master. The details of the authorization procedure are implementation dependent.

Once a device is admitted to the network, it is assigned a unique Device ID. This Device ID is subsequently used as part of the addressing scheme used to allocate transmission opportunities to devices and flows in the Media Access Plan.

The master responds to a REG_REQUEST with a REG_RESPONSE. The response contains a status flag that indicates whether the registration procedure was successful or not. If the procedure is successful, the master downloads network configuration data to the registering device.

If a REGISTRATION_RESPONSE message is not received from the master within the time interval REG_TIMEOUT (T0) period, the registering device should retry after backing off a random amount of time using the RetransmitTimer (see 10.15.6). If the registering device fails to receive a response after MAX_RETRIES, the device should be reinitialized and the sequence restarted.

The Network Admission Protocol is illustrated in the sequence diagram in Figure 10-7.

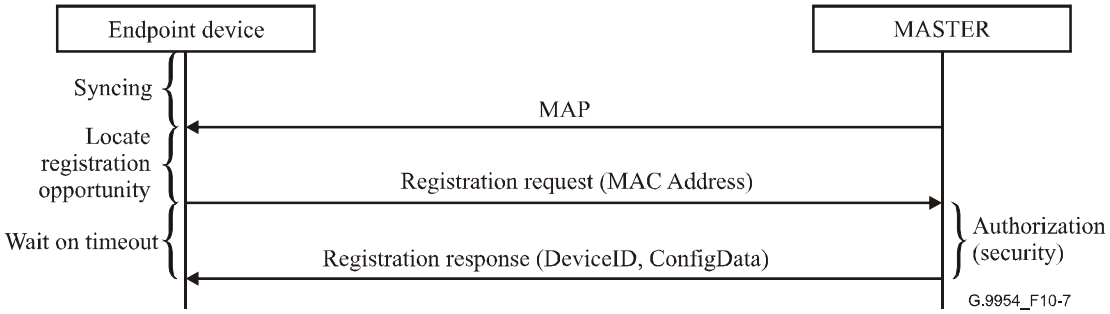


**Figure 10-7/G.9954 – Network admission protocol sequence diagram**

### 10.15.3 Registration state machine

The following state diagram (Figure 10-8) gives a pictorial view of the state transitions in the registration process from the perspective of an endpoint device.

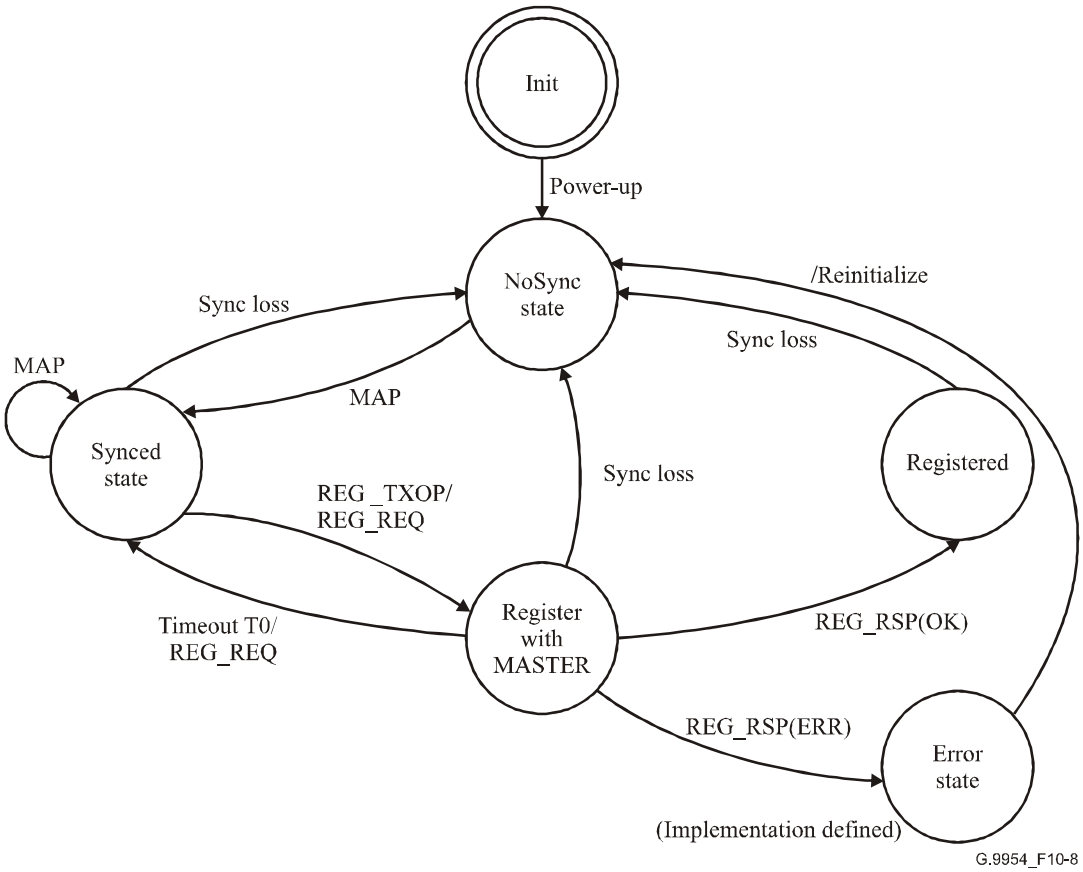


**Figure 10-8/G.9954 – Registration at endpoint device**

The following SDL diagrams (Figures 10-9 and 10-10) provide a complete description of the behaviour of endpoint and master devices during the registration protocol.

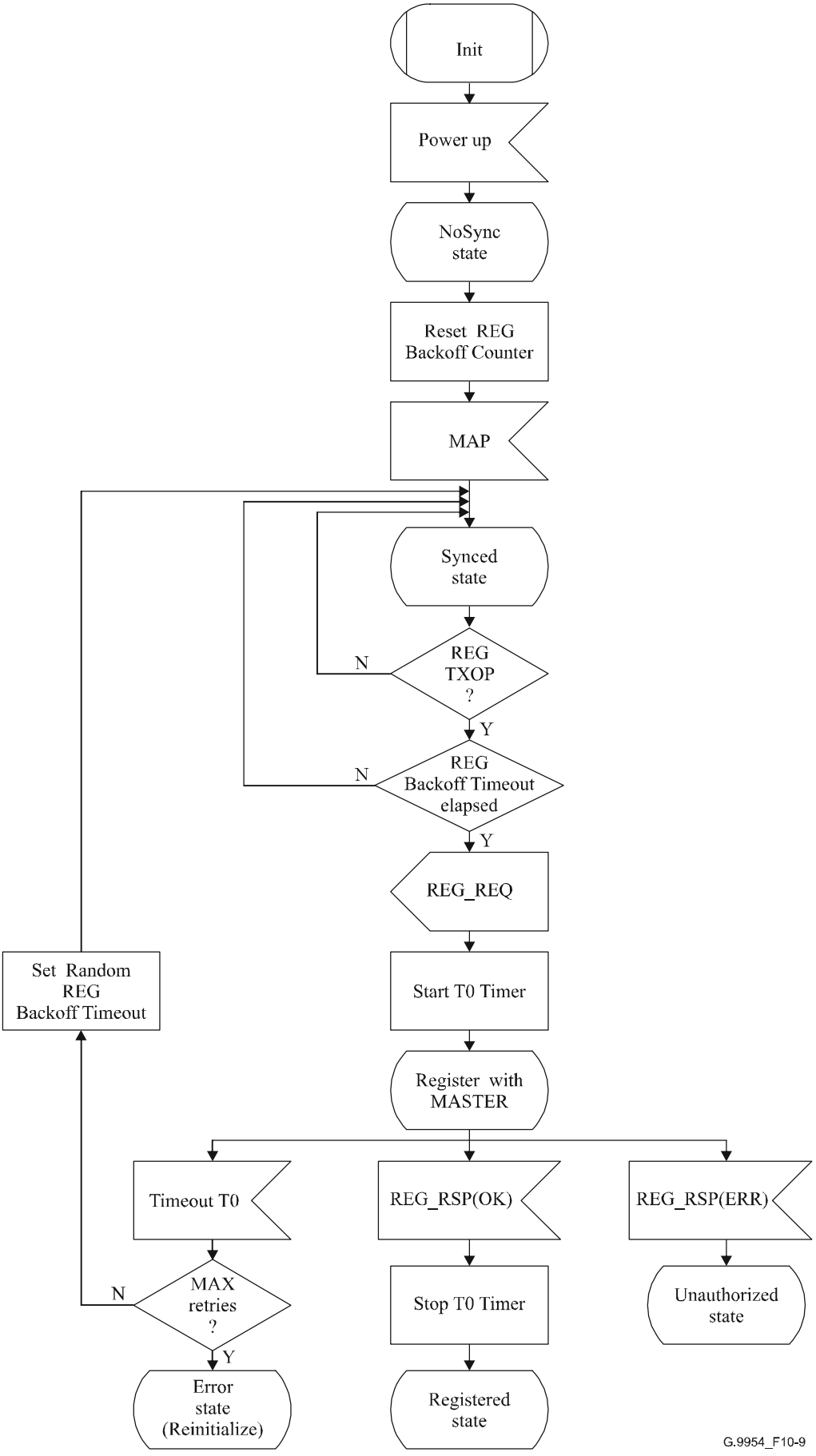


**Figure 10-9/G.9954 – Endpoint registration sequence**

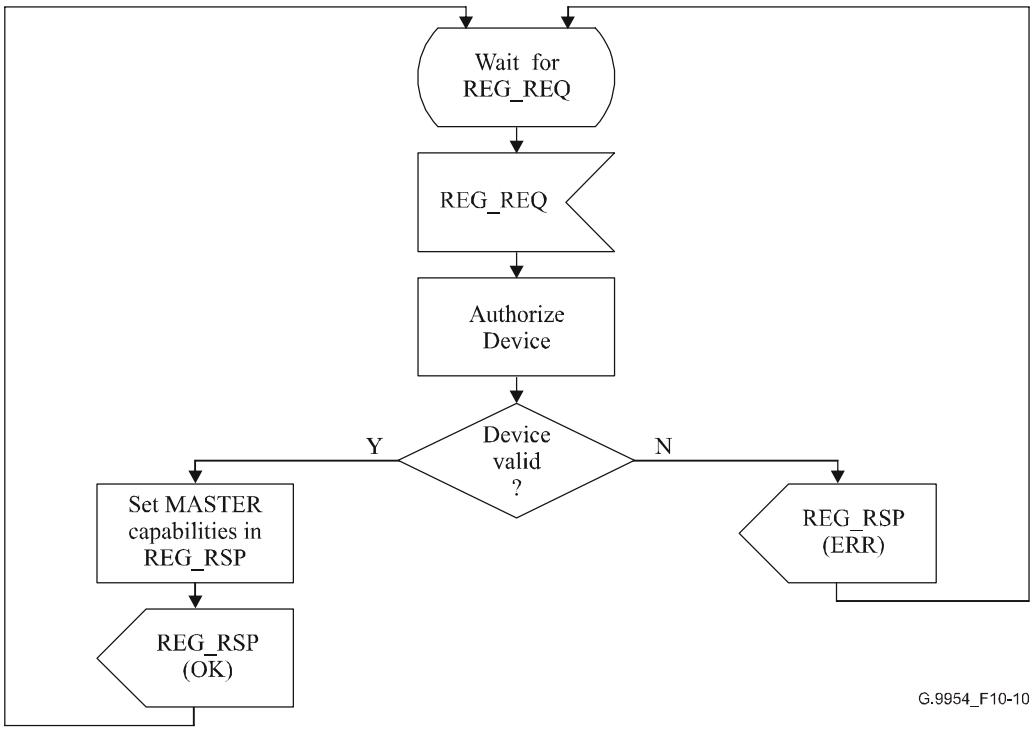


**Figure 10-10/G.9954 – Master registration sequence**

### 10.15.4 Ageing-out registered devices

The master shall maintain an AgeingTimer and at the end of each AgeingTimer period, shall check that a CSA frame was received for each registered device. If a CSA frame was not received for a registered device within the AgeingTimer period, the device shall be de-registered and any associated resources removed.

For a definition of the AgeingTimer, see 10.15.6.1.

### 10.15.5 Frame formats

Registration Control Frames should be sent using Spectral Mask #2, 2 Mbaud, 2 bits per symbol (PE = 33). The format of Registration Control Frames are described in Table 10-55 and Table 10-57.

**Table 10-55/G.9954 – Registration request message**

| Field | Length | Meaning |
|---|---|---|
| DA | 6 octets | Destination Address |
| SA | 6 octets | Source Address of device requesting registration |
| Ethertype | 2 octets | 0x886c (PNT Link Control Frame) |
| LSType | 2 octets | = SUBTYPE_REGISTRATION (32773) |
| LSLength | 2 octets | Number of additional octets in the control header, starting with the LSVersion field and ending with the second (last) octet of the Next Ethertype field. Minimum LSLength is 4 for LSVersion 0. |
| LSVersion | 1 octet | = 0 |
| MsgType | 1 octet | Message type for Registration Request (0) |

**Table 10-55/G.9954 – Registration request message**

| Field | Length | Meaning |
|---|---|---|
| Registration Data | 0-65531 octets | Registration information sent by the device to the master. Includes device capabilities, identification information, etc. Registration Data is optional and TLV encoded. |
| Next Ethertype | 2 octets | = 0 |
| Pad | | Pad to reach minFrameSize if necessary |
| FCS | 4 octets | Frame Check Sequence |

A device generating a Registration message may include the following parameters in the Registration Data.

**Table 10-56/G.9954 – Registration parameters**

| Field | Length | Meaning |
|---|---|---|
| SETag | 1 octet | = 2, Device identity |
| SELength | 1 octet | Total length of TLV extension excluding the tag and length octets (84 octets) |
| Primary_ID | 4 octets | See CERT and DIAG Protocol, in 10.9.6, Table 10-27 |
| Subsystem_ID | 4 octets | See CERT and DIAG Protocol, in 10.9.6, Table 10-27 |
| Vend_Date | 4 octets | See CERT and DIAG Protocol, in 10.9.6, Table 10-27 |
| Manuf_Date | 4 octets | See CERT and DIAG Protocol, in 10.9.6, Table 10-27 |
| Serial_Num | 16 octets | See CERT and DIAG Protocol, in 10.9.6, Table 10-27 |
| Vend_Name | 32 octets | See CERT and DIAG Protocol, in 10.9.6, Table 10-27 |
| Vend_Driver | 16 octets | See CERT and DIAG Protocol, in 10.9.6, Table 10-27 |
| OUI | 3 octets | See CERT and DIAG Protocol, in 10.9.6, Table 10-27 |
| Device_Type | 1 octet | See CERT and DIAG Protocol, in 10.9.6, Table 10-27 |
| Vendor Specific | 1+ octets | Vendor-specific TLV-encoded extension |
| SETag | 1 octet | = 3, Device capabilities |
| SELength | 1 octet | Total length of TLV extension excluding the tag and length octets (3 octets). |
| Max_Flows | 1 octet | Maximum number of flows supported by endpoint device |
| Max Classifiers | 1 octet | Maximum number of classifiers that may be simultaneously installed in the convergence layer |
| Master_Capability | 1 octet | T = Device is capable of becoming master |
| Master Priority | 1 octet | Priority designated to master if device has master capability |
| Vendor Specific | 1+ octets | Vendor-specific TLV encoded extension |

The Registration Response message (Table 10-57) shall be sent by the Master to a device in response to a registration request.

**Table 10-57/G.9954 – Registration response message**

| Field | Length | Meaning |
|---|---|---|
| DA | 6 octets | Destination Address |
| SA | 6 octets | Source Address |
| Ethertype | 2 octets | 0x886c (PNT Link Control Frame) |
| LSType | 2 octets | = SUBTYPE_REGISTRATION (32772) |
| LSLength | 2 octets | Number of additional octets in the control header, starting with the LSVersion field and ending with the second (last) octet of the Next Ethertype field. Minimum LSLength is 6 for LSVersion 0. |
| LSVersion | 1 octet | = 0 |
| MsgType | 1 octet | Message type for Registration Response (1) |
| DeviceID | 1 octet | Device ID assigned to device by master |
| Status | 1 octet | Status of registration request<br>0 OK. Device Registered<br>1 Error |
| Configuration Data | 0-65530 octets | Network configuration information returned by the master upon successful registration of device. This information is optional and TLV encoded. |
| Next Ethertype | 2 octets | = 0 |
| Pad | | Pad to reach minFrameSize if necessary |
| FCS | 4 octets | Frame Check Sequence |

The master responds to a Registration Request with a Registration Response. The following information shall be returned in the Registration Response:

**Status**

A status return code indicating the success or failure of the registration request.

**Device ID**

A device identifier assigned by the master to the device with the specified MAC address.

**Configuration data**

Network configuration data is optional and may be vendor-specific. It may be used to communicate:

- network-wide configuration parameters;
- master capabilities;
- security information;
- service provisioning information.

Table 10-58 describes the values that may appear in the MsgType entry in the Registration Control Frame.

**Table 10-58/G.9954 – MsgType values**

| MsgType | Meaning |
|---|---|
| 0 | Registration Request |
| 1 | Registration Response |
| 2-255 | Reserved |

### 10.15.6 Terms and parameters

- REG_PERIOD – The maximum amount of time between TXOPs that can be used for sending Registration Requests. The value of REG_PERIOD is 50 milliseconds.
- MAX_RETRIES – Number of times an endpoint should retry registration with the master before re-initializing the device. The value of MAX_RETRIES is 5.

#### 10.15.6.1 Timers

- T0 – A one-shot timer set after the transmission of a REG_REQUEST message. Used to time out the expected REG_RESPONSE from the master before retrying the request. This timer is cancelled if a REG_RESPONSE is received.
- RetransmitTimer – A one-shot timer, set to a random interval in the range 1 ms to 1000 ms, inclusive. Used to set the Backoff time before resending a REG_REQUEST in case of a collision during the transmission within a REGISTRATION TXOP. Collisions within UTXOP may be resolved using SMAC collision resolution methods.
- AgeingTimer – Periodic timer with a period of 180 seconds used to determine which registered devices have been actively sending CSA frames.

## 10.16 Master selection protocol

A G.9954 network requires the existence of a network node that takes the role of master in order to coordinate and schedule media transmissions. Although a master is required for a network operating in SMAC mode, not all network nodes necessarily have the functionality to become a master. Amongst those that *do* have the required capabilities, any one of them can potentially become master.

A home network that contains more than one network node that is capable of becoming the master allows for quick recovery from master failure and is inherently more fault/failure tolerant. A master selection protocol shall be used to dynamically select a single master in the presence of multiple potential masters.

The protocol used for discovering and selecting a single master, known as the master Selection Protocol, is described in the following clause.

### 10.16.1 Detection of a managed network

Following power-up, a G.9954 device (configured for G.9954 mode) first tries to detect whether it is operating in a master-controlled network, by listening for MAP control frames and synchronizing with the MAC cycle. If no MAP frames are detected after master_DETECTION_TIMEOUT (T0) interval, the device can conclude that there is no master currently on the network. If the device is master-capable and is willing to become the master, it is able to offer up its candidacy as the network master. If a MAP control frame is received, the device shall synchronize with the advertised MAC cycle and proceed as a regular endpoint device.

### 10.16.2 The master selection procedure

If the network is determined to be unmanaged and a device is capable of and willing to become a master, it can offer up its candidacy by broadcasting a master_SELECTION Control Frame using the asynchronous transmission mode of G.9954. Since several master capable devices may be active on the network at the same time, the master selection procedure includes the mechanism to allow other potential master's to compete for selection as the network master.

Master selection shall be performed according to relative master priority. Each master-capable device shall be assigned a priority using configuration or management parameters. This priority together with the device's MAC address shall be advertised in the master_SELECTION Control Frame. Upon receiving a master_SELECTION Control Frame, G.9954 nodes that are capable themselves of becoming master, may compare the priority of the potential master candidate with its

own assigned priority in order to determine whether it is a "better" candidate. If it is a "better" candidate and it wishes to compete for the role of master, it must broadcast a master_SELECTION Control Frame within master_SELECTION_TIMEOUT (T1) interval.

If an alternative "better" candidacy is not offered within master_SELECTION_TIMEOUT (T1) interval, the master candidate assumes the role of master and may commence the transmission of MAP control frames. If an alternate candidacy is offered, the master with the highest priority shall be assumed to be the master. If there are several candidates with the same priority, the device with the lowest MAC address shall be selected as the master.

All stations that have given up the chance of becoming master should be silent until the selected master's MAP control frame is received.

### 10.16.3 Detection of master failure and recovery

A master is determined to have "failed" if synchronization with the master is lost. Synchronization is lost when a MAP control frame is not received within master_DETECTION_TIMEOUT (T0) interval following the last MAP control frame. Upon detection of master failure, the master may perform an orderly shutdown by inviting a master selection process by sending a master_SELECTION Control Frame with a declared priority of zero.

### 10.16.4 Master selection state machine

Figure 10-11 gives a pictorial view of the state transitions with some minor loss of detail, including omission of events that do not cause state transitions (and have no associated actions), decision logic within a state that leads to the raising of an event and the representation of complex conditions as a high-level "logical" event.

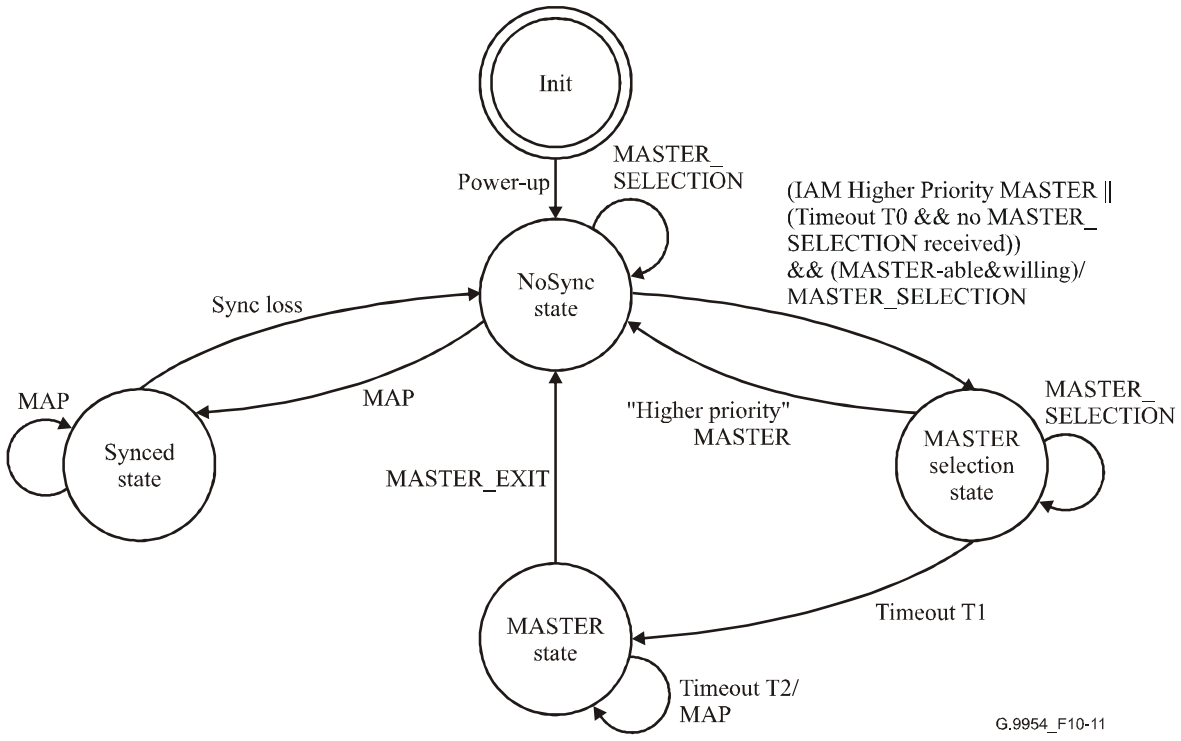


**Figure 10-11/G.9954 – Master-selection state diagram**

Figures 10-12 and 10-13 provide a complete description of the master selection protocol.

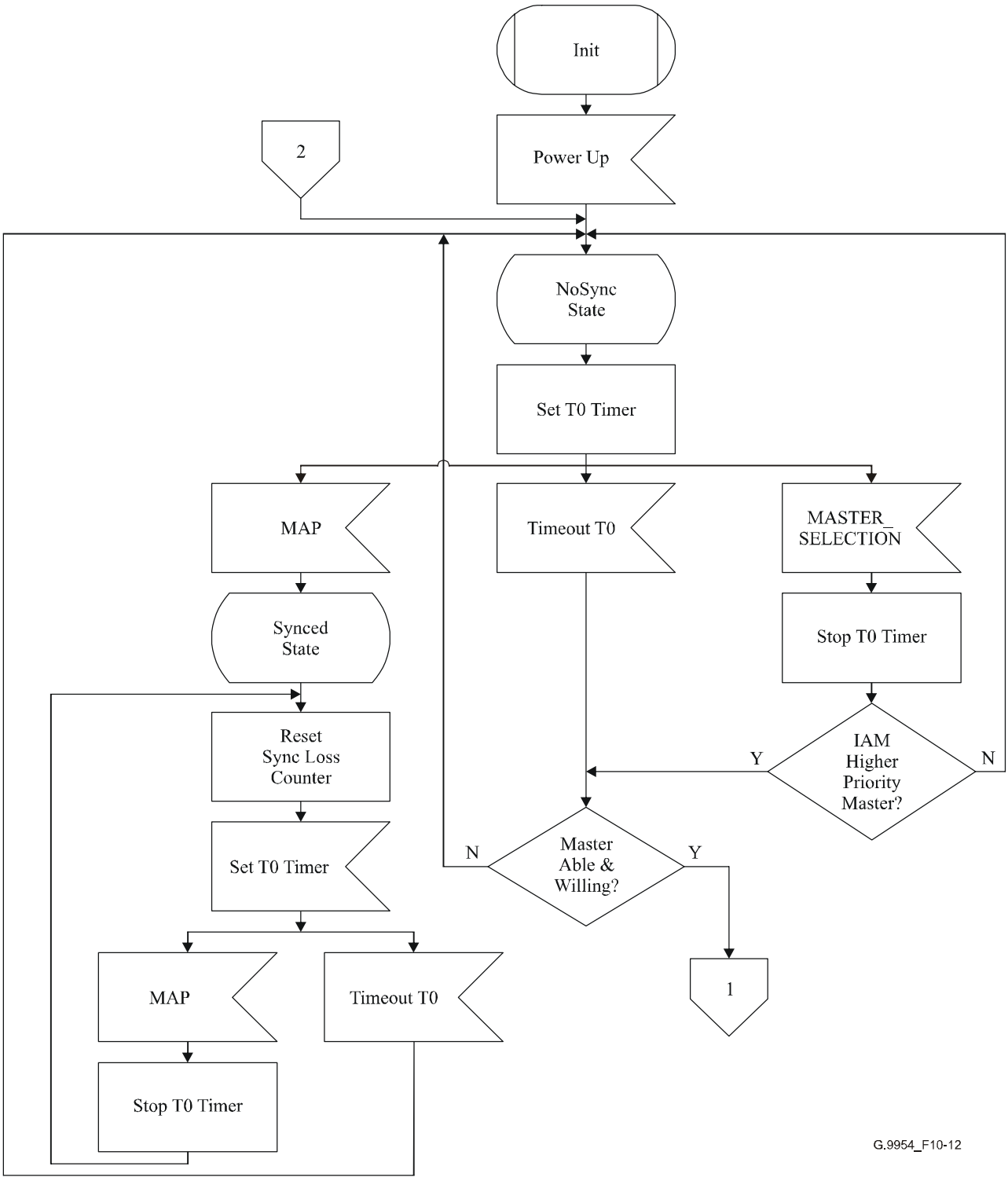


**Figure 10-12/G.9954 – SDL for master selection protocol**

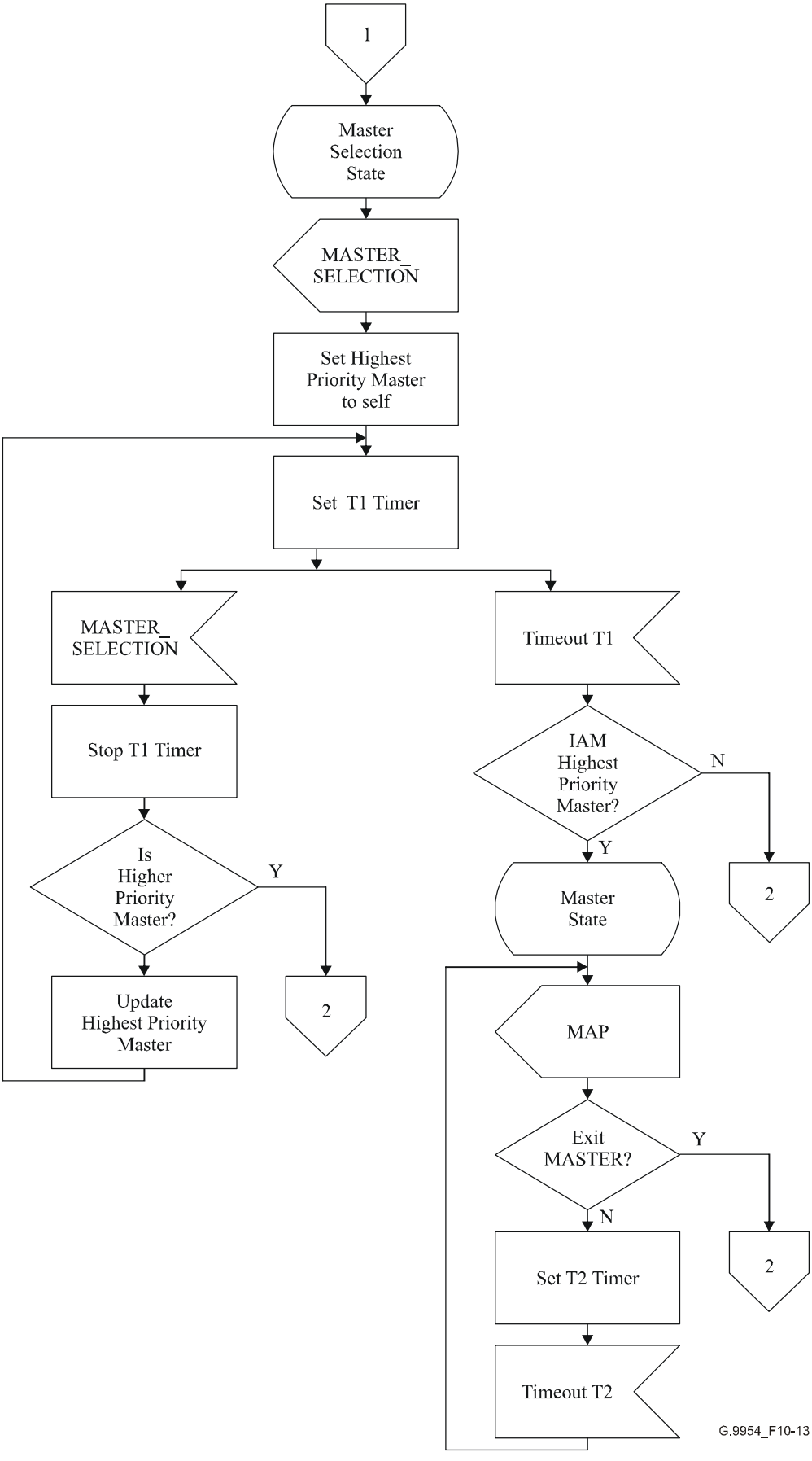


**Figure 10-13/G.9954 – SDL for master selection protocol (cont.)**

### 10.16.5 Master selection protocol messages

**Table 10-59/G.9954 – Master selection control frame**

| Field | Length | Meaning |
|---|---|---|
| DA | 6 octets | Destination Address (broadcast or multicast address) |
| SA | 6 octets | Source Address of the device requesting to become master |
| Ethertype | 2 octets | 0x886c (PNT Link Control Frame) |
| SSType | 1 octet | = SUBTYPE_master_SELECTION (8) |
| SSLength | 1 octet | Number of additional octets in the control header, starting with the SSVersion field and ending with the second (last) octet of the Next Ethertype field. SSLength is 4 for SSVersion 0 |
| SSVersion | 1 octet | = 0 |
| Priority | 1 octet | The assigned master priority. Used to rank potential MASTERs into an order that supports priority selection. Priority values are from 0 to 255 with high numbers representing high priorities. Priority 0 is reserved and can be used by a master to broadcast a desire relinquish master control. |
| Next Ethertype | 2 octets | = 0 |
| Pad | 40 octets | |
| FCS | 4 octets | |

### 10.16.6 Terms and parameters

#### 10.16.6.1 Timers

- T0 – A one-shot timer, set to the value 150 ms, and used to detect the absence of a master on the network. The timer is set by a master-capable device upon entry to the *unsynchronized* state. A device is in *unsynchronized* state when it first wakes up and after MAC Cycle SYNC_LOSS. The timer is cancelled upon the arrival of a MAP control frame. (See 10.14). If the timer T0 expires, a master-capable device may advertise its intention to become master.
- T1 – A one-shot timer set after the transmission or reception of a master SELECTION protocol message. This timer is used to open up a period of time for negotiation between master-capable devices for the role of master. After the T1 timer expires, a master-capable device can decide whether it is the selected master based on its priority and MAC address. The timer T1 and re-armed upon the arrival of a Master-Selection Control Frame. The value of T1 timer is 50 ms.
- T2 – A one-shot timer set by the master to measure the length of the MAC cycle. The value of T2 is variable and dependent upon the scheduler. When the T2 timer expires, the MAP for the next MAC cycle is sent.

## 10.17 Flow Signalling Protocol

The Flow Signalling Protocol is used to dynamically establish and manage service flows with QoS parameters and traffic classification filters defined by upper-layer protocols. More specifically, the Flow Signalling Protocol is used to perform the following flow-related functions:

- Set up a flow and traffic classification filters;
- Modify flow parameters and add or remove classification filters;
- Tear down flows;

- Query QoS parameters for a flow or Class-of-Service.

The Flow Signalling Protocol shall be performed between G.9954 devices at the source and destination of a flow and will be used to establish QoS parameters for the flow. In a master-controlled network, flow signalling shall also be performed between the G.9954 device at the source of the flow and the master, if reserved bandwidth is required. The Flow Signalling Protocol may be initiated by either source or destination devices involved in a unicast flow, or by the source device in a broadcast/multicast flow or by the master.

The Flow Signalling Protocol, in general, involves a 3-way handshake. The handshake allows for negotiation of flow parameters between flow source and destination devices and between flow source and master devices.

Flow signalling with the master is used to reserve media bandwidth to a flow in order to contract QoS throughput, latency/jitter and BER parameters. The master shall be responsible for performing admission control on flow setup requests in order to validate that sufficient media resources exist and the QoS specified by the flow parameters can be met. If the flow is admitted by the master, media transmission opportunities (TXOPs) shall be allocated in the Media Access Plan for the exclusive use of the flow.

In a master-less network, flow signalling may similarly be used to communicate and negotiate flow parameters between source and destination devices. This supports fine-grained rate negotiation at the flow-level for flows with different BER/PER requirements. It also supports a frame burst aggregation scheme that accounts for latency requirements of the flow and memory constraints of the source and destination devices.

The destination of a flow may be a single device identified by a unicast destination address or it may be a group of devices, identified by a broadcast or multicast address. The flow signalling protocol for a group of devices does not require a 3-way handshake in the same manner as a unicast flow setup. Rather, flow parameters are broadcast to the group and no response is required. Group members are always able to initiate an explicit request for flow parameters (from the flow source) for a flow to which they are actively listening or to independently initiate the teardown of an inactive flow.

The remainder of this clause describes the details of the Flow Signalling Protocol and the Flow Signalling Control Frame Formats.

### 10.17.1 Flow Signalling control frames

The SETUP/MODIFY_FLOW_REQUEST control frame (see Table 10-60) is used to request the set up or modification of a flow. The flow being set up or modified is identified using the { FS_SA, FS_DA, FS_FlowID } tuple. The Flow Setup Request is used to set up a flow with a defined set of QoS Flow Parameters. A Flow Modification request is used to modify a QoS Flow Parameter for an existing flow. In both cases, Flow Parameters are always defined for Setup and Modify requests and appear in one of either of two forms as described in 10.17.1.1. Optionally, flow classifiers may be installed at a flow source using the *FlowClassifier* TLV structure (see 10.17.1.2).

**Table 10-60/G.9954 – Set up/modify flow request control frame**

| Field | Length | Meaning |
|---|---|---|
| DA | 6 octets | Destination Address. FS_DA or address of master |
| SA | 6 octets | Source Address |
| Ethertype | 2 octets | 0x886c (PNT Link Control Frame) |
| LSType | 2 octets | = SUBTYPE_FLOW_SIGNALLING (32774) |
| LSLength | 2 octets | Number of additional octets in the control header, starting with the LSVersion field and ending with the second (last) octet of the Next Ethertype field. Minimum LSLength is 58 for SSVersion 0. |
| LSVersion | 1 octet | = 0 |
| MsgType | 1 octet | Message type for SETUP/MODIFY_FLOW_REQUEST (0,3) as defined in Table 10-66 |
| Request_Key | 2 octets | Unique request key used to correlate response/confirm protocol messages |
| FS_SA | 6 octets | MAC address of station at source of flow. Does not necessarily correspond to SA |
| FS_DA | 6 octets | MAC address of station at destination of flow. Does not necessarily correspond to DA |
| FS_FlowID | 1 octet | Unique identifier of the flow between the flow source (FS_SA) and flow destination (FS_DA). The flow identifier is assigned locally by the device at the flow source. If the flow setup request is not initiated by the flow source, the flow identifier shall be specified as NULL. |
| FS_DeviceID | 1 octet | Device ID identifying the device requesting the flow setup or modification. The Device ID is that assigned by the master during the registration process. |
| FlowParameters | 50 octets | QoS properties of flow to be set up. Flow properties are described by a TLV-encoded structure as defined in Table 10-67. |
| FlowClassifiers | N octets | Specification of flow classifiers used to identify a packet belonging to flow. Flow classifiers are optional and described by a TLV-encoded structure as defined in Table 10-67. More than one flow classifier may be defined. |
| Next Ethertype | 2 octets | = 0 |
| Pad | Variable | |
| FCS | 4 octets | Frame Check Sequence |

The SETUP/MODIFY_FLOW_RESPONSE control frame (see Table 10-61) shall be returned in response to a SETUP/MODIFY_FLOW_REQUEST. The response is associated with the corresponding request using the unique *Request Key* assigned by the requestor. The response contains a status indicating whether the request was successful and, in case the requested flow parameters need to be negotiated or modified from their requested values, the modified parameters are returned in a *Flow Parameters* TLV structure.

**Table 10-61/G.9954 – Set up/modify flow response control frame**

| Field | Length | Meaning |
|---|---|---|
| DA | 6 octets | Destination Address |
| SA | 6 octets | Source Address |
| Ethertype | 2 octets | 0x886c (PNT Link Control Frame) |
| LSType | 2 octets | = SUBTYPE_FLOW_SIGNALLING (32774) |
| LSLength | 2 octets | Number of additional octets in the control header, starting with the LSVersion field and ending with the second (last) octet of the Next Ethertype field. Minimum LSLength is 60 for SSVersion 0. |
| LSVersion | 1 octet | = 0 |
| MsgType | 1 octet | Message type for SETUP/MODIFY_FLOW_RESPONSE (1,4) as defined in Table 10-66 |
| Request_Key | 2 octets | Key used to identify the request associated with the response |
| FS_SA | 6 octets | MAC address of station at source of flow. Does not necessarily correspond to SA |
| FS_DA | 6 octets | MAC address of station at destination of flow. Does not necessarily correspond to DA |
| FS_FlowID | 1 octet | Unique identifier of the flow between the flow source (FS_SA) and flow destination (FS_DA). If the flow setup request is not initiated by the flow source, the flow identifier shall be returned in the flow setup response. |
| Status | 1 octet | Status of flow setup request |
| FS_TXOPID | 2 octets | The identifier used to identify TXOPs reserved (allocated) by the master for flow transmissions. This field is assigned only by the master in response to a flow setup request. |
| FlowParameters | N octets | Flow parameters returned in response. The flow parameters returned are those that differ from the corresponding request parameters. Flow parameters are as defined in Table 10-69. |
| Next Ethertype | 2 octets | = 0 |
| Pad | Variable | Pad to reach minFrameSize if necessary |
| FCS | 4 octets | Frame Check Sequence |

The SETUP/MODIFY_FLOW_CONFIRM control frame (see Table 10-62) shall be used to complete the Flow Setup/Modify Protocol. The Flow Setup/Modify sequence is identified by the same *Request_Key* assigned during the request phase of the protocol. The *Confirmation* field is used to indicate acceptance or rejection of the Flow Signalling transaction.

**Table 10-62/G.9954 – Set up/modify flow confirm control frame**

| Field | Length | Meaning |
|---|---|---|
| DA | 6 octets | Destination Address |
| SA | 6 octets | Source Address |
| Ethertype | 2 octets | 0x886c (PNT Link Control Frame) |
| LSType | 2 octets | = SUBTYPE_FLOW_SIGNALLING (32774) |
| LSLength | 2 octets | Number of additional octets in the control header, starting with the LSVersion field and ending with the second (last) octet of the Next Ethertype field. LSLength is 8 for SSVersion 0. |
| LSVersion | 1 octet | = 0 |
| MsgType | 1 octet | Message type for SETUP/MODIFY_FLOW_CONFIRM (2,5) as defined in Table 10-66 |
| Request_Key | 2 octets | Key used to identify the confirmation with request-response sequence |
| FS_SA | 6 octets | MAC address of station at source of flow. Does not necessarily correspond to SA |
| FS_DA | 6 octets | MAC address of station at destination of flow. Does not necessarily correspond to DA |
| FS_FlowID | 1 octet | Flow identifier assigned by the flow source. If a flow setup request is not initiated by the flow source, the flow identifier shall be returned in the flow setup response. |
| Confirmation | 1 octet | Confirmation code for the setup flow protocol sequence |
| FS_TXOPID | 2 octets | The identifier used to identify TXOPs reserved (allocated) by the master for flow transmissions. This field is assigned only by the master in response to a flow setup request. |
| FlowParameters | N octets | Flow parameters, found in the Setup/Modify Flow Response, and requiring re-negotiation. The flow parameters structure is optional and TLV-encoded as described in Table 10-69. |
| Next Ethertype | 2 octets | = 0 |
| Pad | Variable | |
| FCS | 4 octets | Frame Check Sequence |

The FLOW_TEARDOWN_REQUEST control frame shall be used to request the teardown of a flow. The flow is identified by the { FS_SA, FS_DA, FS_FlowID } tuple. The Flow Teardown transaction is ended by the reception of the FLOW_TEARDOWN_RESPONSE control frame (see Tables 10-63 and 10-64).

**Table 10-63/G.9954 – Teardown flow request control frame**

| Field | Length | Meaning |
|---|---|---|
| DA | 6 octets | Destination Address |
| SA | 6 octets | Source Address |
| Ethertype | 2 octets | 0x886c (PNT Link Control Frame) |
| LSType | 2 octets | = SUBTYPE_FLOW_SIGNALLING (32774) |
| LSLength | 2 octets | Number of additional octets in the control header, starting with the LSVersion field and ending with the second (last) octet of the Next Ethertype field. LSLength is 20 for LSVersion 0. |

**Table 10-63/G.9954 – Teardown flow request control frame**

| Field | Length | Meaning |
|---|---|---|
| LSVersion | 1 octet | = 0 |
| MsgType | 1 octet | Message type for TEARDOWN_FLOW_REQUEST(6) as defined in Table 10-66 |
| Request_Key | 2 octets | Key used to identify the teardown request |
| FS_SA | 6 octets | MAC address of station at flow source |
| FS_DA | 6 octets | MAC address of station at flow destination |
| FS_FlowID | 1 octet | ID of flow to be torn down |
| FS_Pad | 1 octet | Ignored on reception |
| Next Ethertype | 2 octets | = 0 |
| Pad | 24 octets | |
| FCS | 4 octets | Frame Check Sequence |

**Table 10-64/G.9954 – Teardown flow response control frame**

| Field | Length | Meaning |
|---|---|---|
| DA | 6 octets | Destination Address |
| SA | 6 octets | Source Address |
| Ethertype | 2 octets | 0x886c (PNT Link Control Frame) |
| LSType | 2 octets | = SUBTYPE_FLOW_SIGNALLING (32774) |
| LSLength | 2 octets | Number of additional octets in the control header, starting with the LSVersion field and ending with the second (last) octet of the Next Ethertype field. LSLength is 8 for LSVersion 0. |
| LSVersion | 1 octet | = 0 |
| MsgType | 1 octet | Message type for TEARDOWN_FLOW_RESPONSE(7) as defined in Table 10-66 |
| Request_Key | 2 octets | Key used to identify the teardown request |
| FS_SA | 6 octets | MAC address of station at flow source |
| FS_DA | 6 octets | MAC address of station at flow destination |
| FS_FlowID | 1 octet | ID of flow to be torn down |
| Status | 1 octet | Status of teardown request |
| Next Ethertype | 2 octets | = 0 |
| Pad | 36 octets | |
| FCS | 4 octets | Frame Check Sequence |

The GET_FLOW_PARAMS_REQUEST control frame shall be used to request the Flow Parameters for a given flow identified by { FS_SA, FS_DA, FS_FlowID }. The Flow Parameters are returned in the GET_FLOW_PARAMS_RESPONSE control frame (see Tables 10-65 and 10-65a).

**Table 10-65/G.9954 – Get flow parameters request control frame**

| Field | Length | Meaning |
|---|---|---|
| DA | 6 octets | Destination Address. FS_SA or address of master |
| SA | 6 octets | Source Address |
| Ethertype | 2 octets | 0x886c (PNT Link Control Frame) |
| LSType | 2 octets | = SUBTYPE_FLOW_SIGNALLING (32774) |
| LSLength | 2 octets | Number of additional octets in the control header, starting with the LSVersion field and ending with the second (last) octet of the Next Ethertype field. Minimum LSLength is 18 for LSVersion 0. |
| LSVersion | 1 octet | = 0 |
| MsgType | 1 octet | Message type for GET_FLOW_PARAMS_REQUEST (8) as defined in Table 10-66. |
| Request_Key | 2 octets | Key used to identify the teardown request |
| FS_SA | 6 octets | MAC address of station at source of flow. Does not necessarily correspond to SA |
| FS_DA | 6 octets | MAC address of station at destination of flow. Does not necessarily correspond to DA |
| FS_FlowID | 1 octet | Identity of flow between FS_SA and FS_DA being queried. |
| FS_pad | 1 octet | Ignored on reception |
| Next Ethertype | 2 octets | = 0 |
| Pad | 0 octet | |
| FCS | 4 octets | Frame Check Sequence |

**Table 10-65a/G.9954 – Get flow parameters response control frame**

| Field | Length | Meaning |
|---|---|---|
| DA | 6 octets | Destination Address. FS_SA or address of master |
| SA | 6 octets | Source Address |
| Ethertype | 2 octets | 0x886c (PNT Link Control Frame) |
| LSType | 2 octets | = SUBTYPE_FLOW_SIGNALLING (32774) |
| LSLength | 2 octets | Number of additional octets in the control header, starting with the LSVersion field and ending with the second (last) octet of the Next Ethertype field. Minimum LSLength is 50 for LSVersion 0. |
| LSVersion | 1 octet | = 0 |
| MsgType | 1 octet | Message type for GET_FLOW_PARAMS_RESPONSE (9) as defined in Table 10-66 |
| Request_Key | 2 octets | Key used to identify the teardown request |
| FS_DA | 6 octets | MAC address of station at destination of flow. Does not necessarily correspond to DA |
| FS_FlowID | 1 octet | Identity of flow between FS_SA and FS_DA being queried |
| Status | 1 octet | Status of the Get Flow Parameters request |

**Table 10-65a/G.9954 – Get flow parameters response control frame**

| Field | Length | Meaning |
|---|---|---|
| FlowProperties | 32 octets | QoS properties of the flow specified in the corresponding request control frame |
| Next Ethertype | 2 octets | = 0 |
| Pad | Variable | |
| FCS | 4 octets | Frame Check Sequence |

Table 10-66 describes the MsgType values used in the Flow Signalling Control Frame.

**Table 10-66/G.9954 – Flow signalling protocol message types**

| MsgType | Meaning |
|---|---|
| 0 | SETUP_FLOW_REQUEST |
| 1 | SETUP_FLOW_RESPONSE |
| 2 | SETUP_FLOW_CONFIRM |
| 3 | MODIFY_FLOW_REQUEST |
| 4 | MODIFY_FLOW_RESPONSE |
| 5 | MODIFY_FLOW_CONFIRM |
| 6 | TEARDOWN_FLOW_REQUEST |
| 7 | TEARDOWN_FLOW_RESPONSE |
| 8 | GET_FLOW_PARAMS_REQUEST |
| 9 | GET_FLOW_PARAMS_RESPONSE |
| 10-127 | Reserved |
| 128-135 | Reserved for master notification of setup, modify and teardown flow request, response and confirm messages |

#### 10.17.1.1 Flow parameters

Flow parameters are specified in the Flow Signalling Control Frames using one of two kinds of TLV encoded structures:

1) Flow Specification structure (see Table 10-67);

2) Flow Parameters structure (see Table 10-68).

The first structure (see Table 10-67), the "Flow Specification" describes each QoS parameter in a flow specification and may be used by a station when setting up a flow or when responding to a GET_FLOW_PARAMS_REQUEST.

**Table 10-67/G.9954 – Flow specification TLV structure**

| Field | Length | Meaning |
|---|---|---|
| SETag | 1 octet | = FS_PARAMS_TAG (2) |
| SELength | 1 octet | Total length of TLV extension excluding the tag and length octets (= 30). |
| Subtype | 2 octets | = Flow Specification(0) |
| ControlWord#1 | 2 octets | See Table 10-69 item 2. |
| ControlWord#2 | 2 octets | See Table 10-69 item 3. |
| PacketSize | 2 octets | See Table 10-69 item 4. |
| MaxPacketSize | 2 octets | See Table 10-69 item 5. |
| MaxDataRate | 2 octets | See Table 10-69 item 6. |
| AvgDataRate | 2 octets | See Table 10-69 item 7. |
| MinDataRate | 2 octets | See Table 10-69 item 8. |
| BER | 1 octet | See Table 10-69 item 9. |
| PE | 1 octet | See Table 10-69 item 10. |
| PacketTimeout | 4 octets | See Table 10-69 item 11. |
| TXTimeslot | 4 octets | See Table 10-69 item 12. |
| FlowTimeout | 4 octets | See Table 10-69 item 13. |

The second structure, the "Flow Parameters" structure, is an incremental structure that can be used to report individual QoS flow parameters or sets of parameters. It shall be used to notify of changes to specific QoS parameters or changes to a specific set of QoS parameters.

**Table 10-68/G.9954 – Flow parameters TLV structure**

| Field | Length | Meaning |
|---|---|---|
| SETag | 1 octet | = FS_PARAMS_TAG (2) |
| SELength | 1 octet | Total length of TLV extension excluding the tag and length octets. Minimum length is 3 and the maximum is 49. |
| Subtype | 2 octets | = Flow Parameters(1) |
| FPType | 1 octet | See Table 10-69. |
| FPLength | 1 octet | See Table 10-69. |
| FlowParameter | 1-4 octets | See Table 10-69. |
| • • • | | [additional instances of Flow Parameters] |

Table 10-69 describes the flow parameters used in the Flow Signalling Control Frames. Diagonal shading is used to show the decomposition of byte and word fields into bit-fields.

**Table 10-69/G.9954 – Flow properties**

| No. | Parameter name | FPType | FPLength [octets] | Values | Comments |
|---|---|---|---|---|---|
| 1 | Pad | 00 | 1 | 0 | |
| 2 | Control Word #1 | 0x01 | 2 | | Control Word is decoded as shown immediately below: |
| | Priority | | Bits 13:15 | 0-7 | Priority assigned to the flow. May be used for G.9951/2 priority semantics |
| | Service Type | | Bits 10:12 | 0-3 | Defines the type of service that the flow supports:<br>0 CBR<br>1 rt-VBR<br>2 nrt-VBR<br>3 BE<br>4~7 Reserved |
| | Max. Latency | | Bits 5:9 | 0-16 | Maximum tolerable transmission and queuing delay according to Table 10-70.<br>17~31 Reserved |
| | Max. Jitter | | Bits 2:4 | 0-3 | Maximum delay variation according to Table 10-71.<br>5~7 Reserved |
| | Reserved | | Bits 0:1 | 0 | Must be set to zero by the transmitter and ignored by the receiver. |
| 3 | Control Word #2 | 0x02 | 2 | | A set of control fields controlling flow behaviour policy. The Control Word is decoded as shown immediately below: |
| | ACK Policy | | Bits 15:15 | 0-1 | 0 None<br>1 LARQ |
| | FEC Policy | | Bits 13:14 | 0-3 | 0 None<br>1 Reed-Solomon<br>2~3 Reserved |
| | Aggregation Policy | | Bits 12:12 | 0-1 | 0 None<br>1 MAC-Level Aggregation |
| | Checksum Error Handling Policy | | Bits 11:11 | 0-1 | 0 Do not discard packets with checksum errors.<br>1 Discard packets with checksum errors.<br>A checksum error includes an error in the FCS or CRC-16 fields of the G.9954 Link Layer Frame or Frame Burst. |
| | Reserved | | Bits 0:10 | 0 | Must be set to zero by the transmitter and ignored by the receiver |

**Table 10-69/G.9954 – Flow properties**

| No. | Parameter name | FPType | FPLength [octets] | Values | Comments |
|---|---|---|---|---|---|
| 4 | Nominal Packet Size | 0x03 | 2 | 0-64 kbit/s | The nominal packet size in octets for packets associated with the service. A value of 0 indicates an unspecified or unknown value. |
| 5 | Max Packet Size | 0x04 | 2 | 0-64 kbit/s | The maximum packet size in octets for packets associated with the service. A value of 0 indicates an unspecified or unknown value.<br>NOTE – Used by the scheduler to ensure that TXOPs are at least large enough to include a single packet. |
| 6 | Max. data rate | 0x05 | 2 | 4 kbit/s – 256 Mbit/s | Peak burst rate in 4 kbit/s units. Takes into account the net (payload) data rate. |
| 7 | Average data rate | 0x06 | 2 | 4 kbit/s – 256 Mbit/s | Average bit rate required by the service in units of 4 kbit/s. |
| 8 | Min. data rate | 0x07 | 2 | 4 kbit/s – 256 Mbit/s | Minimum required bit rate in 4 kbit/s units for the service to operate. This number is expected to be different from zero only for real-time traffic requiring a minimum transmission delay (min ≤ *avg* ≤max). |
| 9 | BER Word | 0x08 | 1 | $10^{-10}$-$10^{-5}$ | Service-level BER in the range $10^{-10} \le BER \le 10^{-5}$.<br>BER is represented by two integer fields: mantissa, m, and exponent, e, such that:<br>$BER = (8 + m) \times 2^{e-43}$<br>When CRC Error Handling Policy is *discard packets with CRC Error*, the BER value is the PER divided by the mean number of bits per packet. For example, suppose the desired $PER = 10^{-2}$ and 1500-byte packets are used, then $BER = 10^{-2}/1200 \approx 10^{-6}$. |
| | Mantissa (m) | | Bits 5:7 | 0-7 | |
| | Exponent (e) | | Bits 0:4 | 7-24 | |
| 10 | PE | 0x09 | 1 | 0-255 | Payload encoding used on logical channel. The value of PE should be derived by Rate Negotiation from BER requirements. |
| 11 | Packet Timeout | 0x0A | 4 | 2^32 – 1 | Amount of time in ms a packet will remain queued before being deleted from the flow queue. A value of 0 indicates that the packet never times out. |

**Table 10-69/G.9954 – Flow properties**

| No. | Parameter name | FPType | FPLength [octets] | Values | Comments |
|---|---|---|---|---|---|
| 12 | TX Timeslot | 0x0B | 4 | 2^32 – 1 | Timeslot of first TXOP defined for the flow. This field can be set by upper layers during flow setup in order to synchronize allocated TXOPs with an external source. This is intended for isochronous services. Time is measured in units of $2^{-13}$ ms and relative to the master's time clock reference. A value of zero indicates the "unknown" value.<br>NOTE – Use of this feature assumes synchronization of the requesting device's clock with the master's clock reference. For further information on Clock Synchronization, see 10.18. |
| 13 | Flow Inactivity Timeout | 0x0C | 4 | 2^32 – 1 | Amount of time in ms a flow will remain "alive" in the absence of any traffic before the flow is automatically torn down and resources released. A value of 0 indicates that the flow is never torn down automatically. |

Tables 10-70 and 10-71 list the possible values for the maximum latency and maximum jitter and their meaning.

**Table 10-70/G.9954 – Maximum latency values**

| Latency | Meaning |
|---|---|
| 0 | No limit |
| 1 | 5 ms |
| 2 | 10 ms |
| 3 | 20 ms |
| 4 | 30 ms |
| 5 | 40 ms |
| 6 | 50 ms |
| 7 | 60 ms |
| 8 | 70 ms |
| 9 | 80 ms |
| 10 | 90 ms |
| 11 | 100 ms |
| 12 | 200 ms |
| 13 | 300 ms |
| 14 | 400 ms |
| 15 | 500 ms |

**Table 10-71/G.9954 – Maximum jitter values**

| Jitter | Meaning |
|---|---|
| 0 | No limit |
| 1 | 5 ms |
| 2 | 10 ms |
| 3 | 20 ms |

### 10.17.1.2 Flow classifier

Flow classifiers are filter specifications that define the criteria by which the G.9954 convergence layer will classify packets and map them to flows. Table 10-72 describes the flow classifier TLV structure used in the SETUP/MODIFY_FLOW_REQUEST Control Frame.

**Table 10-72/G.9954 – Flow classifier data**

| Field | Length | Comments |
|---|---|---|
| SETag | 1 octet | = FS_CLASSIFIER_TAG (Table 10-39) |
| SELength | 1 octet | Total length of TLV extension excluding the tag and length octets |
| Priority | 1 octet | Priority of classifier. Defines order in which classifiers are applied within a Convergence Layer. A higher value indicates a higher priority. |
| Opcode | 1 octet | Classifier action to be applied:<br>0 Add classifier<br>1 Delete classifier |
| ClassifierParam | | Classifier Parameter |
| ClassifierTag | 1 octet | Classifier tag identifier. For a description of classifier tag values, see Table 10-73.<br>Values 0x0E~0xFF are reserved. |
| ClassifierLength | 1 octet | Length of the classifier parameter |
| ClassifierParameter | Variable | A classification parameter whose structure is specific to the ClassifierTag as described in Table 10-73. |

**Table 10-73/G.9954 – Classifier parameters**

| Classifier parameter | Classifier Tag | Length (octets) | Comments |
|---|---|---|---|
| Flow ID | 0x00 | 2 | Flow ID of the flow to which an incoming packet has been determined to belong by higher protocol layers. |
| Destination Address | 0x01 | N * 6 | A list of (N) Ethernet destination addresses |
| Source Address | 0x02 | N * 6 | A list of (N) Ethernet source addresses |
| EtherType | 0x03 | N * 2 | A list of (N) EtherType values. |
| TOS | 0x04 | 3 | IP Type of Service field: ($tos_{low}$, $tos_{high}$, $tos_{mask}$) |
| Protocol | 0x05 | N * 1 | List of protocols: $protocol_1$..$protocol_n$ |
| IP Source Address | 0x06 | N * 8 | A list of source IP (address,mask) tuples |
| IP Destination Address | 0x07 | N * 8 | A list of (N) destination IP (address,mask) tuples |

**Table 10-73/G.9954 – Classifier parameters**

| Classifier parameter | Classifier Tag | Length (octets) | Comments |
|---|---|---|---|
| Source Port Range | 0x08 | N * 4 | A list of (N) source IP port number ranges ($port_{low}$, $port_{high}$)… |
| Destination Port Range | 0x09 | N * 4 | A list of (N) destination IP port number ranges ($port_{low}$, $port_{high}$)… |
| EtherType/802.2 DSAP | 0x0A | N * 1 | LLC DSAP Address |
| EtherType/802.2 SSAP | 0x0B | N * 1 | LLC SSAP Address |
| User Priority | 0x0C | 2 | A range of 802.1D user priority values $pri_{low}$, $pri_{high}$ |
| VLAN ID | 0x0D | 2 | The 802.1Q VLAN identifier. Only the leftmost 12 bits are significant. |

### 10.17.2 Flow signalling transactions

Multiple flow signalling transactions may be initiated by a station simultaneously using a uniquely assigned *Request Key*. All protocol messages belonging to the same transaction shall use the same *Request Key*. The *Request Key* shall be assigned by the initiator of the flow signalling transaction.

### 10.17.3 Flow Signalling Protocol Sequences

#### 10.17.3.1 Flow Setup Protocol Sequence

Flow setup shall be performed between source and destination endpoints of a flow using the *Flow Setup Protocol Sequence*. Either the source or destination stations may initiate the flow setup.

The purpose of flow setup signalling is to establish a set of well-defined and negotiated flow parameters between flow endpoints.

If reserved bandwidth (QoS Contracts) is required for a flow, and the network is operating in SMAC mode, the master shall be informed of the flow setup parameters, by the flow source, after flow parameters have been negotiated. Notification of flow setup to the master and reservation of bandwidth shall be performed using the same 3-way handshake for flow setup used between two endpoint nodes.

The master may also be the source or destination endpoint of a flow. This is a special case of the standard Flow Setup Protocol Sequence.

NOTE – When the master is at the endpoint of a flow, no further master notification is required in order to reserve bandwidth beyond the original flow setup signalling.

The different flow setup protocol sequences are defined in the following clauses.

##### 10.17.3.1.1 Source-initiated flow setup procedure

To set up a flow between two G.9954 devices on the network where the device initiating the flow setup shall be the device at the source of the flow, the initiator shall send a SETUP_FLOW_REQUEST message to the device at the flow destination. The SETUP_FLOW_REQUEST message shall contain a *Request Key* assigned by the initiator and identifying the flow setup transaction, the flow identity and flow QoS parameters. The flow identity shall be locally assigned by the initiator by assigning a Flow Identifier that will be unique within the context of the flow *source* and *destination* addresses.

After sending the SETUP_FLOW_REQUEST, the station shall set a timer and wait for up to FLOW_RESPONSE_TIMEOUT (T1) ms for a SETUP_FLOW_RESPONSE message. If no

response is received within the timeout period, the request shall be resent using the same *Request Key*. This process shall be performed until the MAX_FLOW_SIGNALLING_RETRIES.

Upon receiving a SETUP_FLOW_REQUEST, the destination station shall set up the flow locally. It may offer suggested modifications to the flow parameters in order to better suit the flow to the endpoint's resource restrictions. Any modified parameters shall be returned in the SETUP_FLOW_RESPONSE. After sending a SETUP_FLOW_RESPONSE, the destination endpoint shall start a timer and wait for up to FLOW_CONFIRM_TIMEOUT (T2) ms for the SETUP_FLOW_CONFIRM message. If a SETUP_FLOW_CONFIRM is not received within this timeout period, a SETUP_FLOW_RESPONSE message shall be retransmitted. This procedure continues MAX_FLOW_SIGNALLING_RETRY times before the destination shall abandon the flow setup operation and close the transaction.

If a SETUP_FLOW_RESPONSE is received, the station shall disable the timer (T1) and check the returned status and flow parameters. If flow parameters were modified by the destination station in its response, then the source shall adjust its flow parameters accordingly. If the return status in the SETUP_FLOW_RESPONSE is OK and the modified parameters are acceptable to the source, the source-station shall return a FLOW_SETUP_CONFIRM message with a status of OK and the flow setup transaction closed. If the offered flow parameters are rejected by the source station, it shall return a confirmation code of REJECT together with the rejected parameters.

Upon receiving a SETUP_FLOW_CONFIRM message the station shall disable the timer (T2). If the *Confirmation Code in the* SETUP_FLOW_CONFIRM is OK, then the destination station may complete the flow setup transaction. If the *Confirmation Code* is REJECT, the destination station may either end the flow set up transaction or it may modify its offer using the same FLOW_SETUP_RESPONSE/CONFIRM cycle. If the flow cannot be successfully set up, a SETUP_FLOW_RESPONSE status of ERROR should be returned and the flow setup transaction shall be closed at source and destination.

If a flow is not successfully set up between source and destination stations, flow data shall be sent using AMAC transmission rules and using channel parameters defined for the logical channel between Source and Destination Address. If the network is master-controlled, the transmission shall be performed within a contention TXOP.

If a flow is successfully set up and the network is master-controlled, bandwidth may be reserved for the flow by signalling the flow setup with the master. For more information on reserved bandwidth allocation for a flow, see 10.17.3.1.4.

Figure 10-14 illustrates the *Flow-Setup Signalling Protocol* used to set up a flow between devices A (the source) and device B (the destination) when the initiator of the *Flow-Setup* transaction in the example is Device A. This example illustrates the timer periods (T1, T2) used in the flow signalling protocol as well as Rate Negotiation (RRCF) performed over the flow channel.

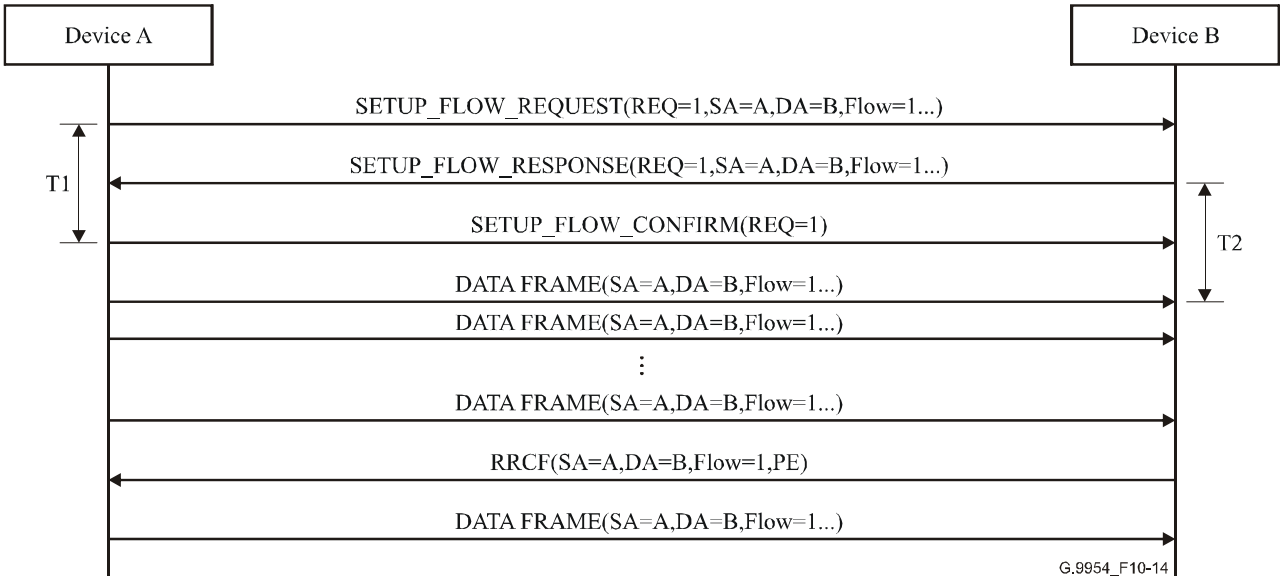


**Figure 10-14/G.9954 – Source-initiated flow setup procedure**

### 10.17.3.1.2 Destination-initiated flow setup procedure

Flow setup, when initiated by the flow destination, is similar to the procedure described in 10.17.3.1.1. The difference between the sequences is as follows:

The Flow_ID specified in the FLOW_SETUP_REQUEST is NULL since the Flow_ID must be defined by the station at the source of the flow. The assigned Flow_ID is returned in the FLOW_SETUP_RESPONSE.

Flow parameter negotiation proceeds as for the case of the source-initiated flow setup.

Figure 10-15 illustrates the *Flow-Setup Signalling Protocol* used to set up a flow between devices A (the source) and device B (the destination) when the initiator of the *Flow-Setup* transaction in the example is Device B.

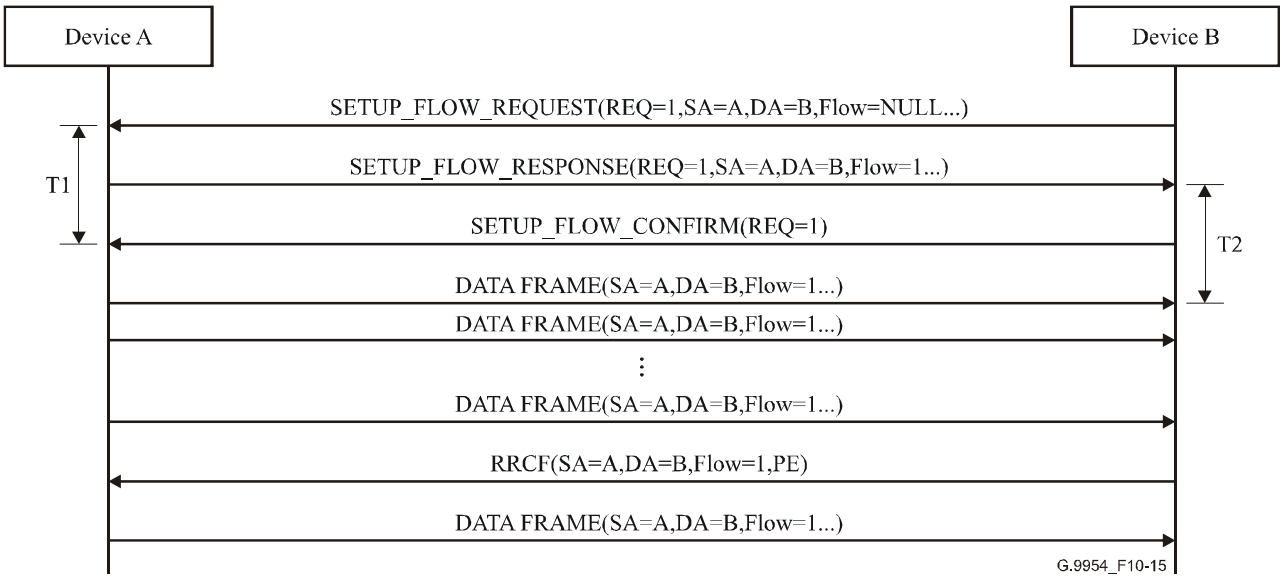


**Figure 10-15/G.9954 – Destination-initiated flow setup protocol**

#### 10.17.3.1.3 Broadcast/Multicast flow setup procedure

When setting up a broadcast/multicast flow, the *Flow-Setup Signalling Protocol* does not use the standard 3-way handshake to set up the flow since the initiator of the flow setup cannot wait for a response from all broadcast/multicast group members. Rather, flow setup shall be signalled by broadcasting the SETUP_FLOW_REQUEST without waiting for a response and without having to reply with a confirm. Flow parameters (except for Payload Encoding (PE)) cannot be negotiated for Broadcast/Multicast flows. Payload Encoding shall be negotiated using the standard Rate Negotiation mechanism for broadcast/multicast channels as described in 10.4.

In order to allow a broadcast/multicast group member to acquire flow parameters at any time, in case the SETUP_FLOW_REQUEST was not received, or the broadcast/multicast group member came alive after the establishment of the flow, a station may make a request to *Get Flow Parameters* at any time using the GET_FLOW_PARAMS_REQUEST. The request is sent to the station at the flow source. The station at the flow source, upon receiving a GET_FLOW_PARAMS_REQUEST shall return the parameters for the designated flow using the GET_FLOW_PARAMS_RESPONSE message.

The Flow Setup Protocol Sequence in the case of broadcast/multicast flows is illustrated in Figure 10-16.

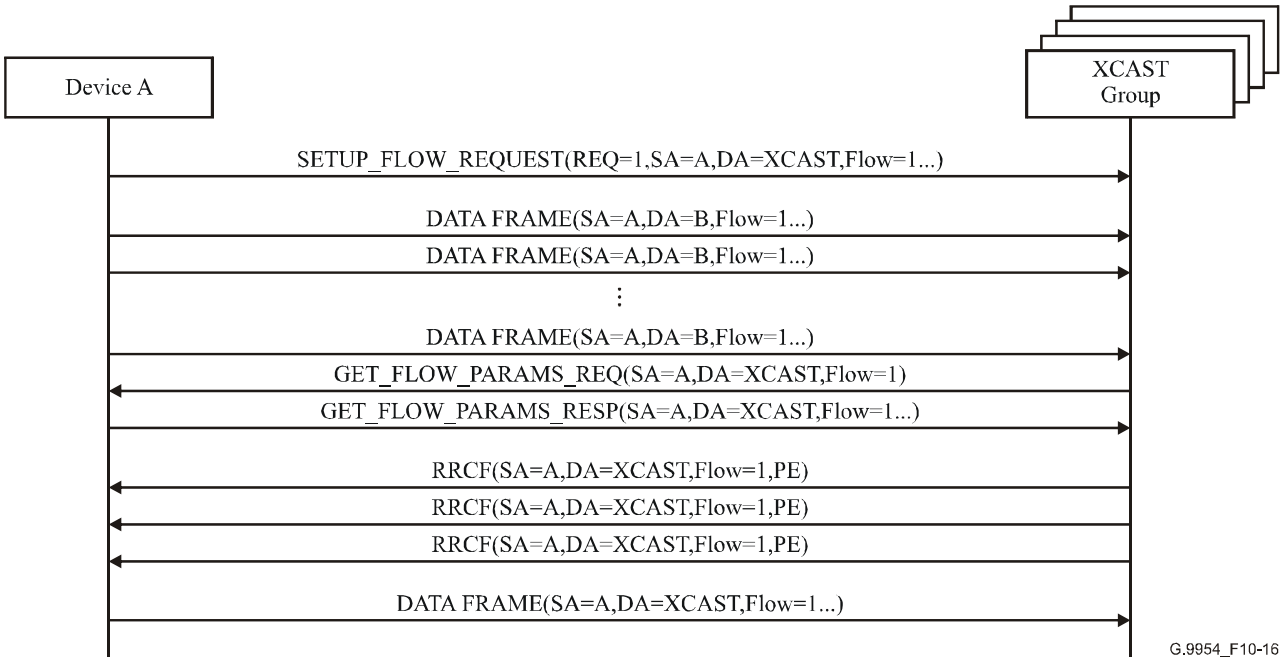


**Figure 10-16/G.9954 – Multicast flow setup**

#### 10.17.3.1.4 Master flow setup notification procedure

As described in 10.17.3.1.1, 10.17.3.1.2 and 10.17.3.1.3, the flow setup protocol shall be performed between flow source and destination devices, irrespective of whether the network is master-controlled or not. This allows the definition of flows with defined latency, rate and BER characteristics. This information may be used by transmitter and receiver devices to negotiate appropriate channel parameters for the flow (e.g., buffer requirements, payload encoding, etc.).

If the network is master-controlled, explicit TXOPs may be reserved for an established flow by signalling the master of the flow setup using the regular flow setup signalling protocol.

To signal flow setup with the master, the protocol shall be initiated by the flow source. The same 3-way protocol handshake shall be used as for the regular flow setup operations between source and destination devices. If the flow is admitted by the master, reserved TXOPs shall be allocated by the

master and assigned in the advertised master-generated MAP. The TXOPs shall be allocated by the master scheduler in such a manner and position so as to provide sufficient bandwidth and meet latency and jitter requirements defined for the flow in flow parameters.

Devices at the source of a flow shall be *registered* with the master in order to be able to request reserved bandwidth.

Figure 10-17 illustrates the *Flow Setup Protocol Sequence* including master Flow Setup Notification. The *Flow Setup Protocol Sequence*, appearing between Device A and Device B (i.e., within the double-sided arrow), represents the protocol sequence as described in Figures 10-14, 10-15 and 10-16. The *Flow Setup Protocol Sequence* between Device A and master represents the reserved bandwidth allocation request.

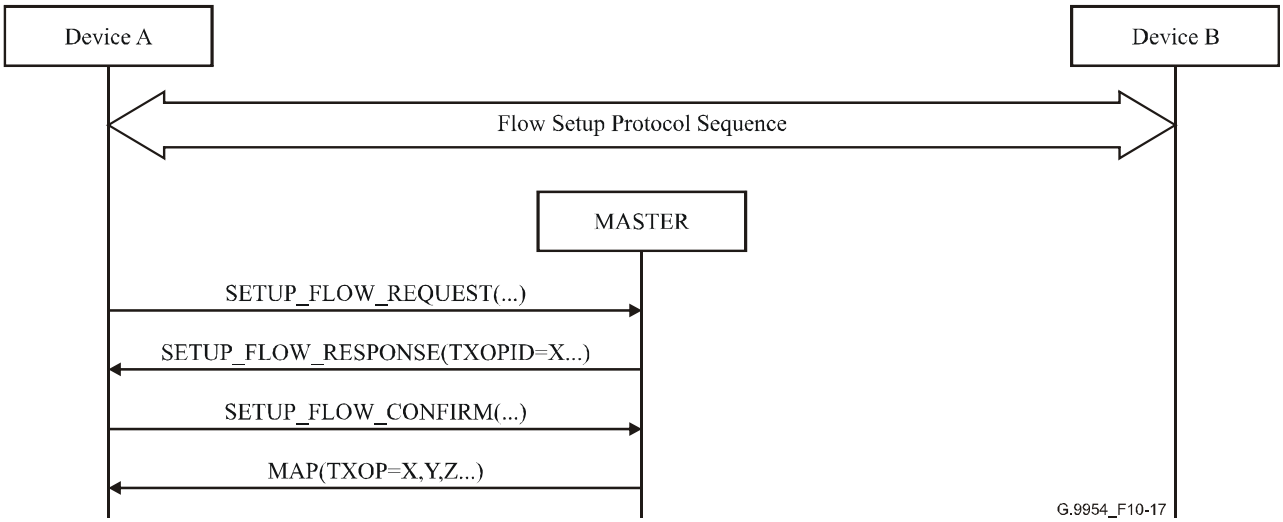


**Figure 10-17/G.9954 – Master flow setup notification**

#### 10.17.3.1.5 Master-initiated and terminated flow setup procedure

If the device initiating the Flow Setup Sequence is the master, the flow setup sequence proceeds normally, as for the case of a regular endpoint station (see 10.17.3.1.1 and 10.17.3.1.2). In this case, admission control may be performed by the master before the protocol sequence begins. Furthermore, the master need not be notified of the flow setup in order to reserve bandwidth. This shall be performed automatically by the master for flows requiring reserved bandwidth.

Similarly, for flows whose endpoint terminates at the master, the *Flow Setup Protocol Sequence* proceeds as for the regular case and bandwidth reservation shall be performed automatically by the master as required.

NOTE – Bandwidth for a flow need not be allocated immediately by the master and may be deferred until flow channel parameters (e.g., Payload Encoding) have been determined.

### 10.17.3.2 Flow Modification Protocol Sequence

The Flow Modification Protocol Sequence closely follows the *Flow Setup Protocol*. It similarly involves a 3-way REQUEST-RESPONSE-CONFIRM protocol exchange sequence between flow source and destination devices and optionally between flow source and master device.

Flow modification can be initiated by flow source or destination devices. Similar to the flow setup protocol, the master shall be informed of modifications to flows for which bandwidth has been explicitly reserved, if the modified parameters effect bandwidth reservation.

Modifications to the following parameters effect master bandwidth reservation:

– Data Rate (Minimum, Average, Maximum);

– Maximum Latency/Jitter;

– Payload Encoding;

– Nominal Packet Size.

#### 10.17.3.2.1 Flow modification procedure

The device requesting the flow modification shall open a flow signalling transaction and send a MODIFY_FLOW_REQUEST message containing a specification of the flow parameters to be modified and/or optionally the traffic classification filters to be installed in the device at the source of the flow.

After sending the MODIFY_FLOW_REQUEST, the initiator shall set a timer and wait for up to FLOW_RESPONSE_TIMEOUT (T1) ms for a MODIFY_FLOW_RESPONSE. If the timer expires before the response is received, the MODIFY_FLOW_REQUEST shall be resent up to MAX_FLOW_SIGNALLING_RETRY times before the flow modification request shall be abandoned.

Upon receiving a MODIFY_FLOW_REQUEST message, the receiving device should look up the specified flow in its list of established flows and, if found, set up a new flow signalling transaction. Modified parameters should be checked and, if acceptable, the flow parameters should be updated accordingly. A MODIFY_FLOW_RESPONSE with a *Status* of OK should subsequently be returned within (T1)/2 ms from the time the MODIFY_FLOW_REQUEST was received. If the modified flow parameters are unacceptable, a MODIFY_FLOW_RESPONSE with a *Status* of REJECT should be returned. The rejected parameters should be returned in the response message.

The remainder of the protocol sequence, including renegotiation of flow parameters (if necessary), and the termination of the flow signalling transaction proceeds as for the case of *Flow Setup*. This is illustrated in Figure 10-18.

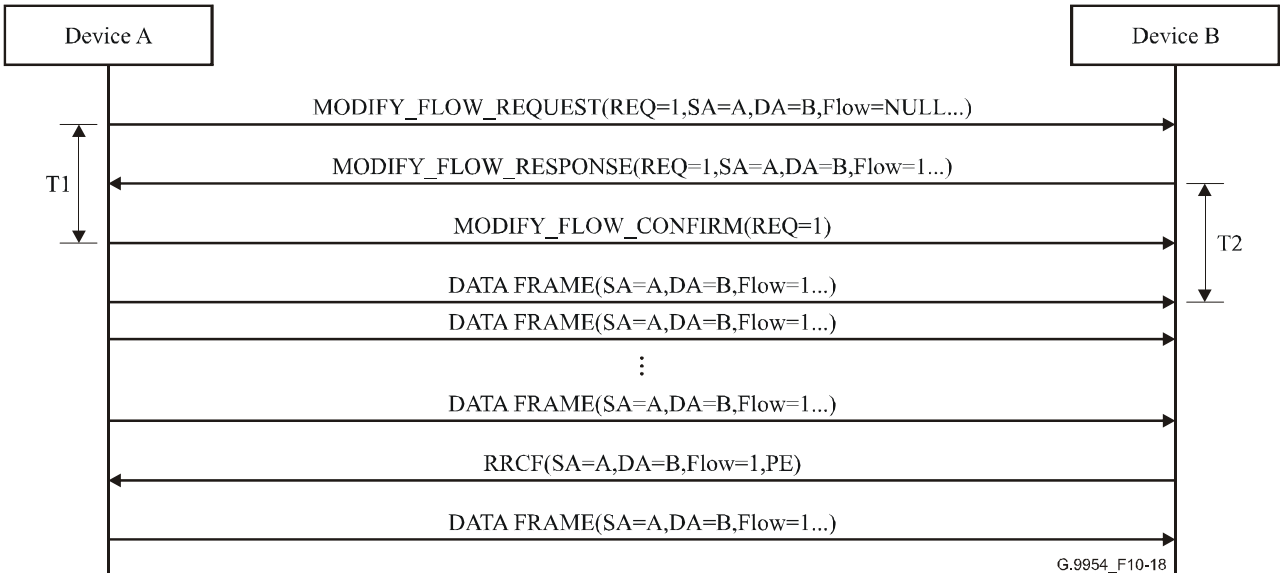


**Figure 10-18/G.9954 – Modify flow signalling protocol**

#### 10.17.3.2.2 Master-notification and flow-modification

If a flow that has bandwidth reserved by the master is modified, the master shall be notified of any modifications to flow parameters that effect bandwidth allocation. Notification shall be performed using the *Modify Flow Signalling Protocol*.

Flow parameters that may be modified and effect bandwidth reservation, are as defined in 10.17.3.2.

The *Modify Flow Signalling Protocol* between the device at the flow source and the master is the same as described in 10.17.3.2.1.

### 10.17.3.3 Flow teardown protocol sequence

Flows are torn down using the *Flow Teardown Protocol Sequence.* A flow may be torn down in response to an explicit request from an upper protocol layer or after a flow-parameter configurable period of inactivity (see *Flow Inactivity Timeout* flow parameter in 10.17.1.1).

The flow teardown sequence is normally initiated by the device at the source of the flow after sensing a period of flow inactivity greater than or equal to the flow's *Flow Inactivity Timeout.* Flow teardown may also be initiated by the device at the destination of a flow if it senses a period of inactivity greater than its *Flow Inactivity Timeout* parameter.

The *Flow Teardown Protocol Sequence* involves a REQUEST-RESPONSE message sequence. The initiator shall identify the flow by *Source Address, Destination Address* and *Flow ID.* When a flow is torn down, the resources it binds shall be released.

If a flow that has bandwidth reserved to it by the master is torn down, the master shall be notified by the device initiating the *Flow Teardown Sequence.*

If a registered device is no longer detected, as indicated by the absence of Capability and Status Announcement Control Frames (CSA), the master shall de-register the device and teardown all flows sourced at the device. Similarly, devices at the source of a flow shall detect the absence (using CSA timeout) of a device at the flow's destination and shall teardown such flows accordingly.

The *Flow Inactivity Timeout* at the source of a flow shall be greater than the *Flow Inactivity Timeout* at a flow's destination in order to eliminate flow teardown race conditions.

NOTE – The initiator of a *Flow Setup Protocol Sequence* should guarantee this above requirement by specifying the desired Flow Inactivity Timeout accordingly. This means that for a Flow Setup initiated by the flow source, the Flow Inactivity Timeout specified in the setup request should be forced to be greater than the parameter at the flow source. Similarly, for a flow setup initiated by the flow destination, the Flow Inactivity Timeout specified should be less than the value used at flow destination.

#### 10.17.3.3.1 Station-Initiated Flow Teardown Procedure

*Flow Teardown Protocol* signalling shall be performed between devices found at the endpoints of a flow or between the device at the source of a flow and the master. In either case, a device initiates the *Flow Teardown Protocol Sequence* by sending a TEARDOWN_FLOW REQUEST message containing the identity of the flow to be torn down and a unique *Request Key* identifying the flow signalling transaction. The initiating device shall subsequently set a timer and wait for up to FLOW_RESPONSE_TIMEOUT (T1) ms for a TEARDOWN_FLOW_RESPONSE message before resending the teardown request. This procedure shall be performed up to MAX_FLOW_SIGNALLING_RETRY times before the flow teardown transaction shall be terminated and the flow torn down locally.

A device receiving a TEARDOWN_FLOW_REQUEST shall search for the identified flow in its database of active flows and, if found, the device should tear down the flow locally and release resources bound to the flow. In all cases, a TEARDOWN_FLOW_RESPONSE should be returned within (T1)/2 ms.

The *Flow Teardown Protocol Sequence* is illustrated in Figure 10-19. The scenario described shows a flow teardown sequence between devices at the endpoints of a flow and between the device at the flow source and the master.

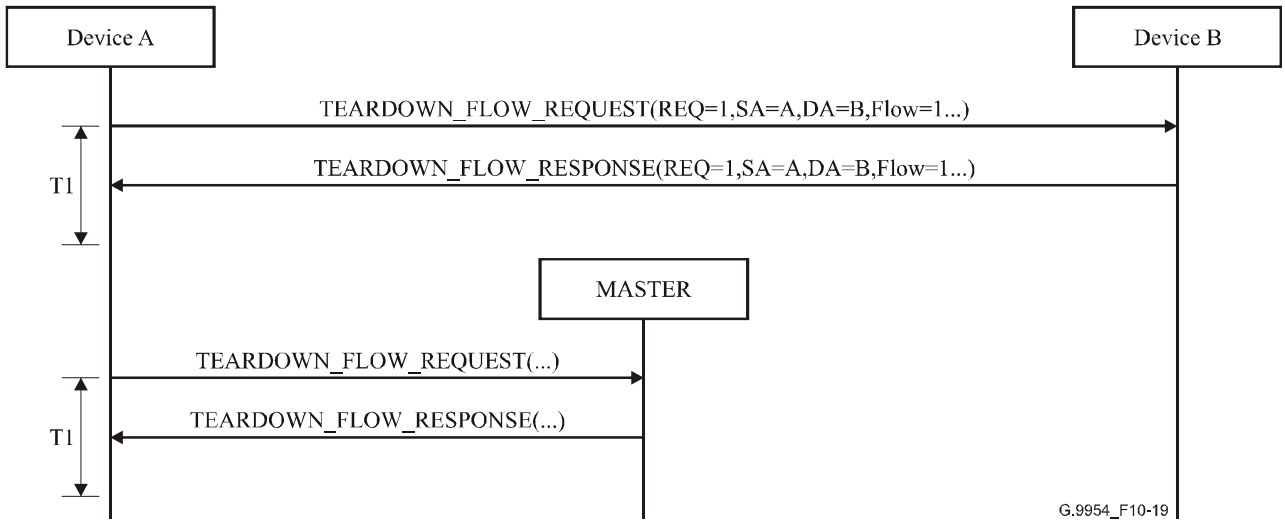


**Figure 10-19/G.9954 – Flow teardown protocol**

#### 10.17.3.3.2 Flow teardown signalling with the master

If a flow has bandwidth reserved to it by the master and the flow is torn down, the master shall be notified by the device found at the flow source. The master shall be notified using the *Flow Teardown Protocol Sequence*, the same as for devices found at the endpoints of a flow.

#### 10.17.3.3.3 Broadcast and multicast flow teardown

To tear down a broadcast or multicast flow, the TEARDOWN_FLOW_REQUEST shall be sent by the device at the source of the flow. The TEARDOWN_FLOW_REQUEST message shall be sent using the broadcast/multicast address. The initiating device shall not wait for a TEARDOWN_FLOW_RESPONSE and may end the transaction after sending the teardown request.

If a broadcast/multicast group member does not receive the TEARDOWN_FLOW_REQUEST, the flow shall be timed out by each device using the standard *Flow Inactivity Timeout* mechanism.

### 10.18 Timestamp Report Indication message (optional)

Synchronization to a master clock reference may be required, by some endpoint devices, in order to synchronize sampling rates or to synchronize the allocation of media TXOPs with an external source.

To support synchronization with a master clock, a master clock reference device distributes its clock to all devices on the network.

Any device on the network may be a master clock reference to some group of clock slave devices. More than one master clock reference device may co-reside on the network simultaneously. Typically a clock slave device should synchronize to a single master clock reference. There is no requirement that the device acting at the master for SMAC mode be a master clock reference.

The Timestamp Reporting mechanism assumes the ability of a master clock reference to latch the transmission timestamp of a well-known message (the Timestamp Report message itself) and to send the latched timestamp value in the subsequent Timestamp Report Indication message. Furthermore, it assumes the ability of an endpoint device to latch the receive timestamp of the same message. The time difference between the latched receive time at the endpoint and the latched transmit time at the master clock reference is used to adjust the clock at the endpoint to compensate for the calculated clock frequency error. This is illustrated in Figure 10-20.

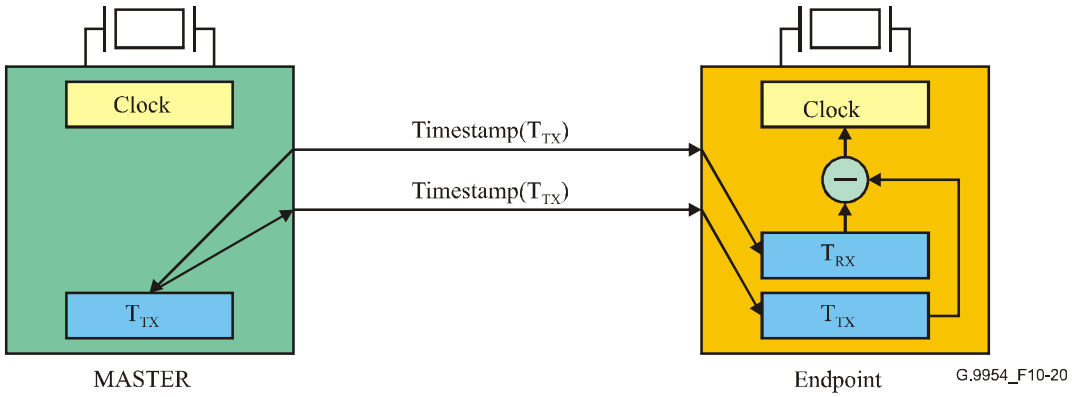


**Figure 10-20/G.9954 – Timestamp Report Indication**

The master clock reference may transmit a Timestamp Report Indication at any time. It should transmit pairs of these indications in successive frames. For each Timestamp Report Indication message transmitted, the master clock reference shall increment the Timestamp Sequence Number by one. The Timestamp Sequence Number may start at any arbitrary value.

In measuring the start-of-transmission and start-of-reception times by the master clock reference and endpoint, respectively, the measurements must be defined with respect to a common point in the frame. That point is immediately following the MAC-layer Source Address field. A particular implementation may make its actual measurement with respect to other points in the frame, but in following the procedures below, it must correct the measured value so that the time corresponds to the specified point.

All endpoints that require data sampling synchronization are encouraged to receive the Timestamp Report Indication and measure the start-of-reception time for received frames that contain this message. On reception of a Timestamp Report Indication, the endpoint shall perform the following actions:

- Record the start-of-reception time of the current frame along with the Timestamp Sequence Number and Timestamp from the received Timestamp Report Indication.
- Compare the Timestamp Sequence Number parameter contained in the current frame with that of the most recently received Timestamp Report Indication. If the timestamps have a modulo difference of one, then continue. Otherwise, stop processing the message at this point.
- Calculate the relative frequency error of its internal clock by the following:

$$\text{Frequency error} = [(R_{(seqnum-1)} - R_{(seqnum-2)})/(C_{seqnum} - C_{(seqnum-1)})] - 1$$

where:

$R_{(seqnum-1)}$ is the start-of-reception time of the frame containing the Timestamp Report Indication with the previous sequence number, as measured by the endpoint's local clock.

$R_{(seqnum-2)}$ is the start-of-reception time of the frame containing the Timestamp Report Indication with the sequence number two less (modulo) than that of the current frame, as measured by the endpoint's local clock.

$C_{seqnum}$ is the Timestamp value indicated in the Timestamp Report Indication in the current frame (which corresponds to the start-of-transmission time of the frame containing the Timestamp Report Indication with the previous sequence number, as measured by the master).

$C_{(seqnum-1)}$ is the Timestamp value indicated in the Timestamp Report Indication with the previous sequence number (which corresponds to the start-of-transmission time of the frame containing the Timestamp Report Indication with the sequence number two less (modulo) than that of the current frame, as measured by the master clock reference).

- Adjust the local clock according to the determined frequency error using a locally defined algorithm.

The mechanism that the master clock reference or endpoints use to measure the frame start-of-transmit and start-of-receive time, respectively, is locally defined.

### 10.18.1 Timestamp Report Indication frame format

**Table 10-74/G.9954 – Timestamp Report Indication frame format**

| Field | Length | Meaning |
|---|---|---|
| DA | 6 octets | Destination Address = FF:FF:FF:FF:FF:FF |
| SA | 6 octets | Source Address is that of the master clock reference. |
| Ethertype | 2 octets | 0x886c (PNT Link Control Frame) |
| SSType | 1 octet | = SUBTYPE_TIMESTAMP_REPORT (8) |
| SSLength | 1 octet | Number of additional octets in the control header, starting with the SSVersion field and ending with the second (last) octet of the Next Ethertype field. SSLength value is 8 for SSVersion 0 |
| SSVersion | 1 octet | = 0 |
| Reserved | 1 octet | Set to zero by sender and ignored by the receiver |
| TimestampSequenceNr | 2 octets | A sequence number that increments by one each time a Timestamp Report Indication is transmitted |
| Timestamp | 4 octets | The time measured by the master of the start-of-transmission of the previous frame containing the Timestamp Report Indication. The time is measured in units of ticks clocked at the frequency defined by ClockFrequency. |
| ClockFrequency | 4 octets | Frequency of the clock used to clock the timestamp reference expressed in kHz. E.g., 8192 kHz for an 8.192 MHz clock with resolution of $2^{-13}$ ms. |
| Next Ethertype | 2 octets | = 0 |
| Pad | 36 octets | |
| FCS | 4 octets | |

# Annex A

# Mechanical interface (MDI)

## A.1 MDI connector

The wire connector mounted on the PNT device shall be an RJ11 female connector with the pin assignment of Table A.1.

**Table A.1/G.9954 – RJ11 MDI connector pin assignment**

| Contact | Signal |
|---|---|
| 1 | Not used |
| 2 | Not used |
| 3 | TX/RX (+) |
| 4 | TX/RX (–) |
| 5 | Not used |
| 6 | Not used |

A depiction of the connector is shown in Figure A.1. The two pins labeled TX/RX(+) and TX/RX(–) constitute the PNT W1 interface to the phonewire network.

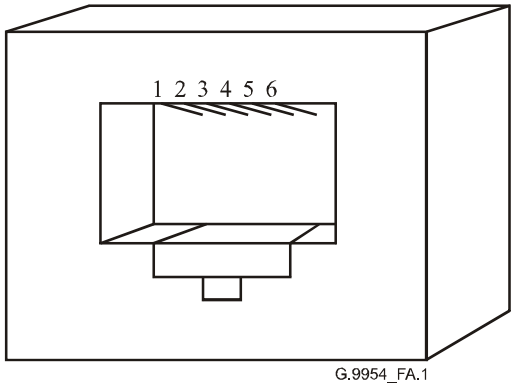


**Figure A.1/G.9954 – RJ11 female wire connector**

# Annex B

# Network test loops

Ten test loops are defined for evaluating the performance of PNT receivers. This annex includes specification of the wire types and the topologies.

## B.1 Wire model

The wire labeled "quad" in the following diagrams is assumed to be Belden 1242A, or wire with equivalent primary parameters. The wire labeled "flat" is assumed to be Mouser flat 4-wire 26-AWG cable (stock number 172-UL4210), or wire with equivalent primary parameters. All other wire types are Belden UTP-5 of the specified gauge.

For simulations, the "BT #1" [1] model is used to generate primary parameters *R*, *L*, *G*, and *C* vs. frequency. This model is given as:

$$R(f) = \sqrt[4]{r_o^4 + a \cdot f^2}$$

$$L(f) = \frac{l_0 + l_\infty \cdot \left(\frac{f}{f_m}\right)^b}{1 + \left(\frac{f}{f_m}\right)^b}$$

$$G(f) = g_0 \cdot f^{g_e}$$

$$C(f) = c_\infty + \frac{c_0}{f^{c_e}}$$

The parameter set for each of the wire types used in the next clause is given in Table B.1. The assumption is that *R*(*f*) is in units of ohms/mi., *L*(*f*) is in units of mH/mi., *G*(*f*) is in units of µMhos/mi., and *C*(*f*) is in units of µF/mi.

**Table B.1/G.9954 – Model parameters for wires**

| Model parameter | Belden 1242A quad | Mouser flat 4-wire | Belden UTP-5 (24AWG) |
|---|---|---|---|
| $r_0$ | 406.65 | 643.4 | 277.2 |
| $A$ | 0.2643 | 0.757 | 0.278 |
| $l_0$ | 1.229 | 1.27 | 0.9863 |
| $B$ | 0.794 | 0.654 | 0.83 |
| $l_\infty$ | 0.927 | 0.953 | 0.718 |
| $f_m$ | 386e3 | 697e3 | 500e3 |
| $g_0$ | 0.0432 | 0.519 | 0.000282 |
| $g_e$ | 0.8805 | 0.7523 | 0.869 |
| $c_0$ | 0.121 | 0.04 | 0 |
| $c_\infty$ | 0.071 | 0.06875 | 0.083 |
| $c_e$ | 0.245 | 0.122 | 0 |

## B.2 Test loops

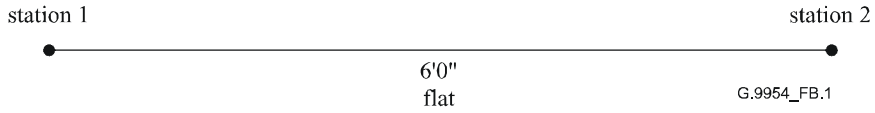


**Figure B.1/G.9954 – Test loop No. 1**

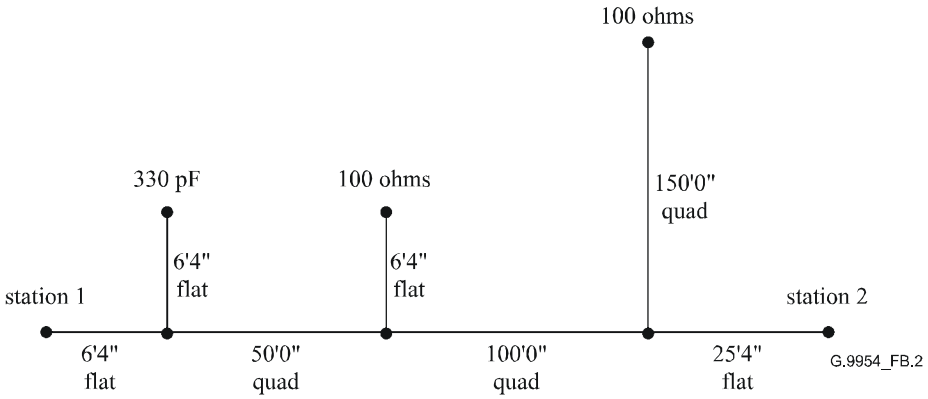


**Figure B.2/G.9954 – Test loop No. 2**

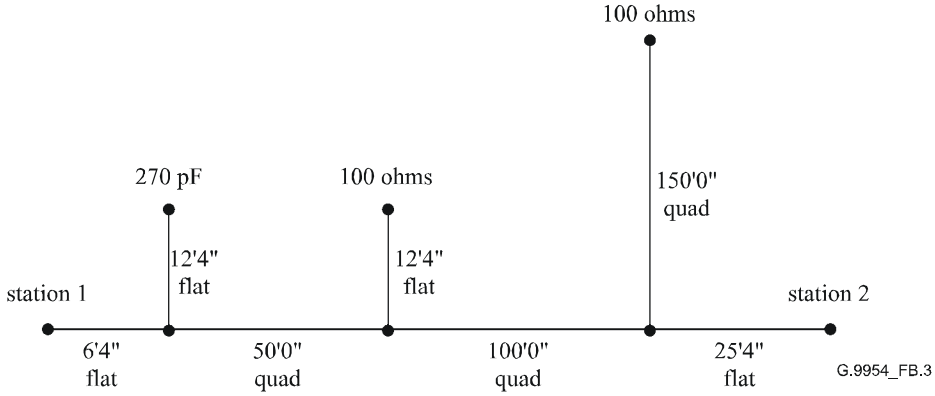


**Figure B.3/G.9954 – Test loop No. 3**

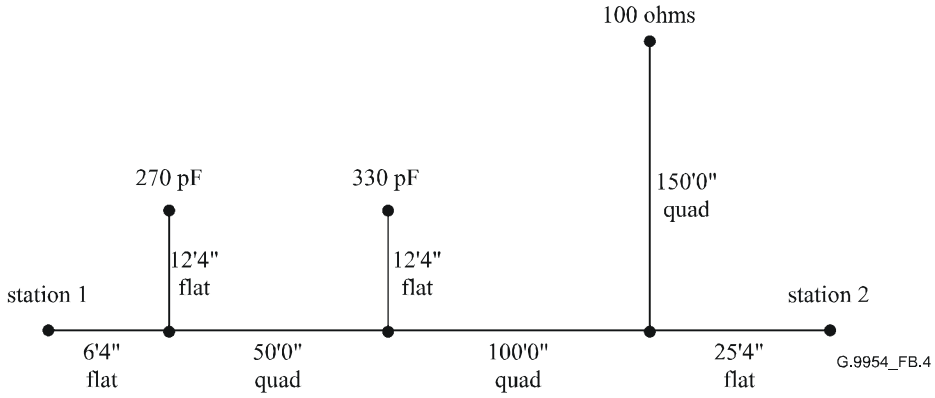


**Figure B.4/G.9954 – Test loop No. 4**

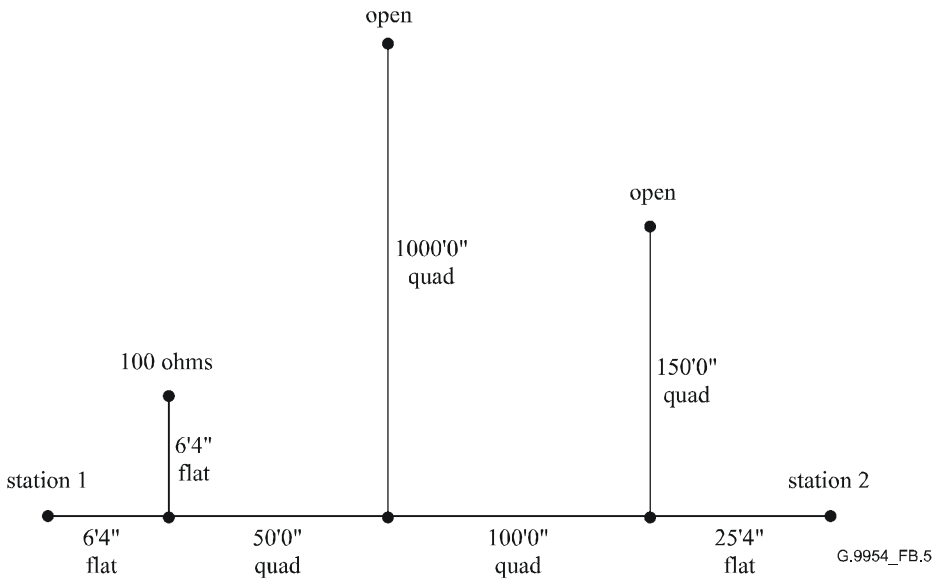


**Figure B.5/G.9954 – Test loop No. 5**

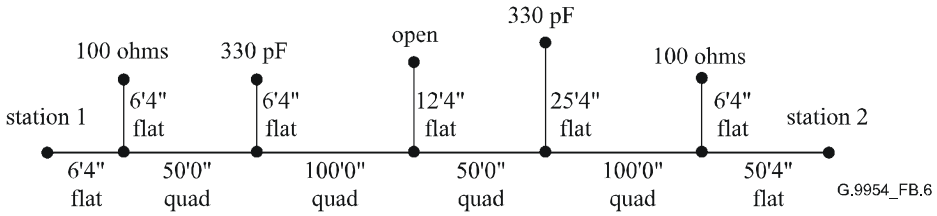


**Figure B.6/G.9954 – Test loop No. 6**

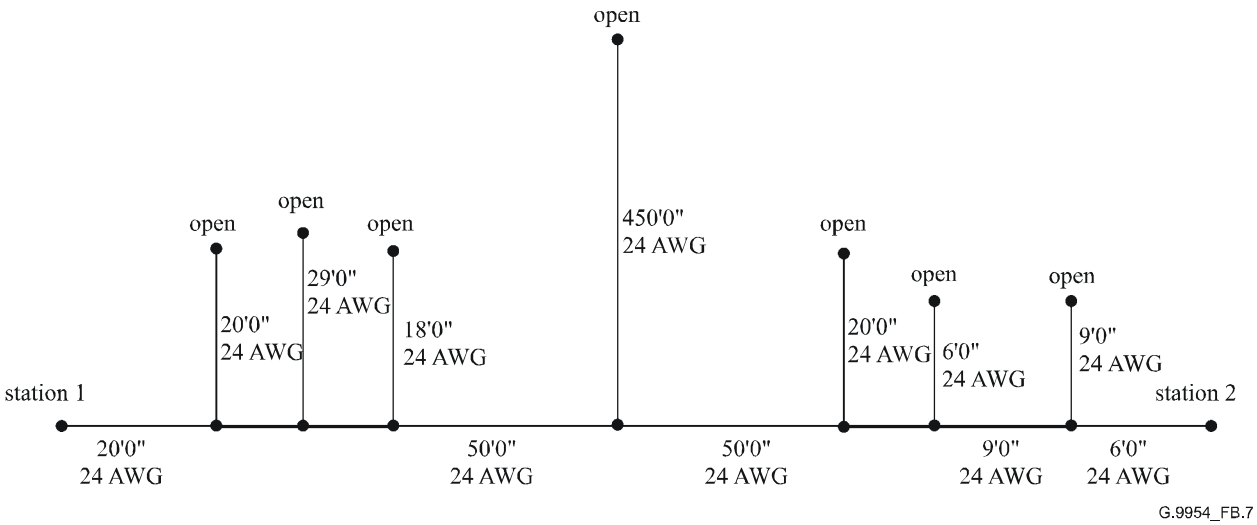


**Figure B.7/G.9954 – Test loop No. 7**

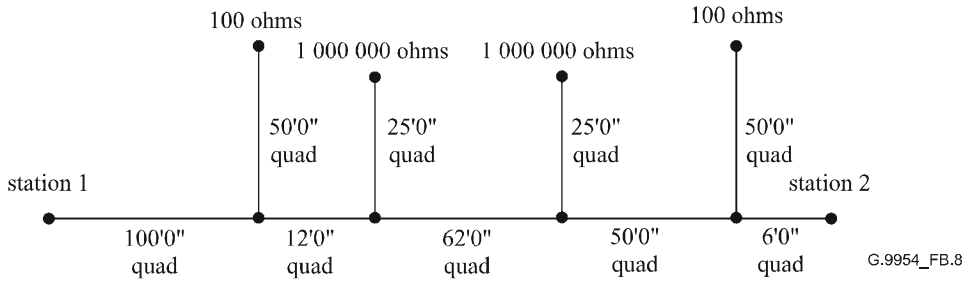


**Figure B.8/G.9954 – Test loop No. 8**

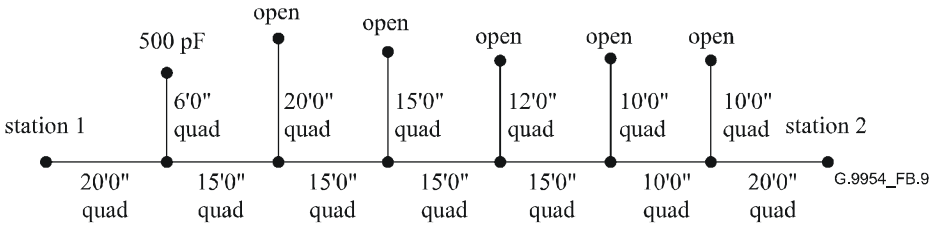


**Figure B.9/G.9954 – Test loop No. 9**

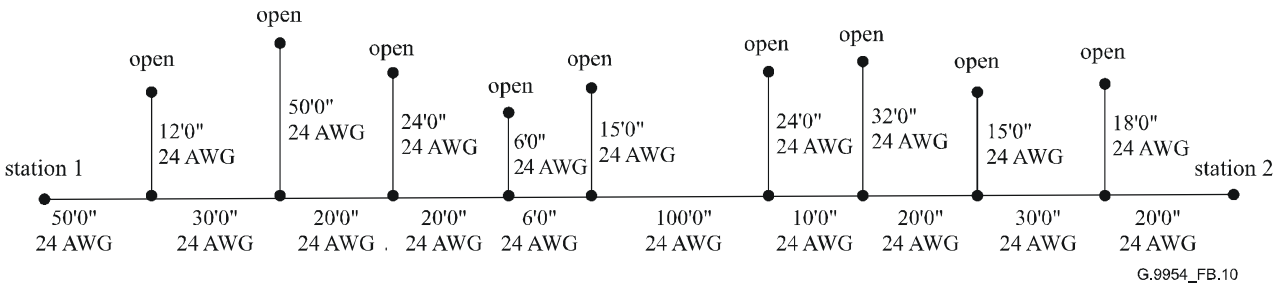


**Figure B.10/G.9954 – Test loop No. 10**

# Appendix I

# Convergence layers

The Convergence Layer is a protocol-specific sub-layer that maps various transport layer protocols into the native primitives of the LLC sub-layer. The LLC sub-layer provides a protocol-independent interface and a well-defined QoS framework. It is the responsibility of the Convergence Layer to translate the native protocol into this underlying framework.

This Appendix describes the G.9954 Convergence Layer, its logical interfaces and general requirements for particular protocol-specific Convergence Layers. Since the logical interface between Convergence and Link Layers are between protocol stack layers developed by the same vendor, there is no issue of interoperability between different vendor solutions. Consequently, the content of this Appendix should be considered informative in nature and used only as a guideline for implementations.

## I.1 Overview

The G.9954 protocol stack supports interfaces and bridging to external network protocols through the Convergence Layer. The protocol convergence sub-layers available on a G.9954 device are advertised using the Link Layer Capability and Status Announcement protocol (see 10.6). By default, Ethernet and IP convergence layers are defined.

It is the responsibility of the protocol Convergence Layer to map data packets arriving from a particular interface onto the *flows* appropriate for the particular service. Flows defined for a particular Convergence Layer are set up by the Convergence Layer itself in an implementation-dependent way, possibly during initialization, on receipt of data from upper layers, on network admission or upon demand. The flow traffic and rate parameters for a flow may be also defined in an implementation-dependent way, perhaps by upper-layer protocols, or configured using management operations or configuration data held in non-volatile storage.

G.9954 Convergence sub-layers considered for the G.9954 protocol stack include the IEEE 802.3/Ethernet, IP protocols, USB and IEEE 1394. In addition, bridging interfaces to broadband access protocols, such as DOCSIS and wireless access protocols, such as IEEE 802.11 and IEEE 802.16, are envisioned, as are application-level convergence sub-layers for applications such as VoHPNA and for the delivery of MPEG transport streams.

Protocol mapping and convergence at a well-defined level of the protocol stack enables a degree of synchronization between external and home protocols. Furthermore, given QoS defined in terms that are similar to those of the external network, this further supports the extension of QoS from external networks into the home network.

The Convergence Layer may perform the following functions:

- Interface to higher layer protocols and receive PDU from the upper layers.
- Signal the setup of traffic flows and classifiers in local and peer MAC, Link-Layer and Convergence Layer entities.
- Classify upper-layer PDUs, using built-in knowledge of the protocols, and map the PDUs to underlying flows.
- Perform address bridging and translation functions.
- Perform any special PDU processing before passing them onto the Link/MAC layers (e.g., removal of payload header information).
- Send upper-layer PDUs to PNT Link/MAC layers.

- Receive PDUs transported by the PNT PHY/MAC layers and performs any protocol specific processing before delivery to upper protocol layers.
- Perform peer-to-peer convergence sub-layer signalling.
- Perform data sampling and synchronization control.

No assumptions should be made as to the system partitioning of Link and Convergence Layer functions as it is possible to implement both of these protocols both on-chip or in external host drivers.

## I.2 Convergence Layer primitives

The following clause describes the Convergence Layer interface to the lower layers of the G.9954 protocol stack. Since the details of the Convergence Layer-LLC interface are implementation dependent, this interface is described in terms of a set of primitives supported by the Link-Layer Control Service Access Point (LLC_SAP).

The following primitive types are defined:

– req (request) – primitive used by the convergence sub-layer to request a service from the LLC sub-layer.

– cnf (confirm) – primitive used by the LLC sub-layer to confirm that a requested activity has been completed.

– ind (indication) – primitive used by the LLC sub-layer to notify the convergence sub-layer of any specific service related activity.

– rsp (response) – primitive used by the convergence sub-layer to acknowledge the receipt of an indication primitive from the LLC sub-layer.

The primitives and their relationships are illustrated in Figure I.1:

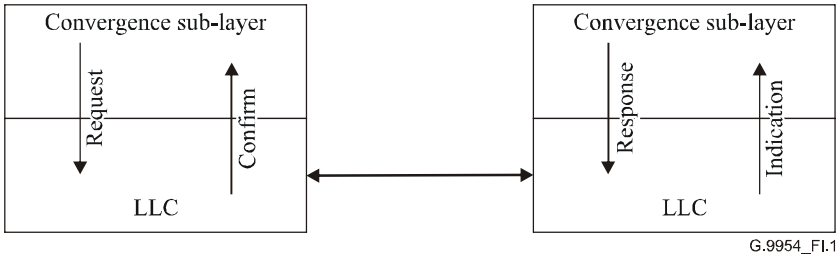


**Figure I.1/G.9954 – Service primitives**

Figure I.2 illustrates the Convergence Layer – Link-Layer interface.

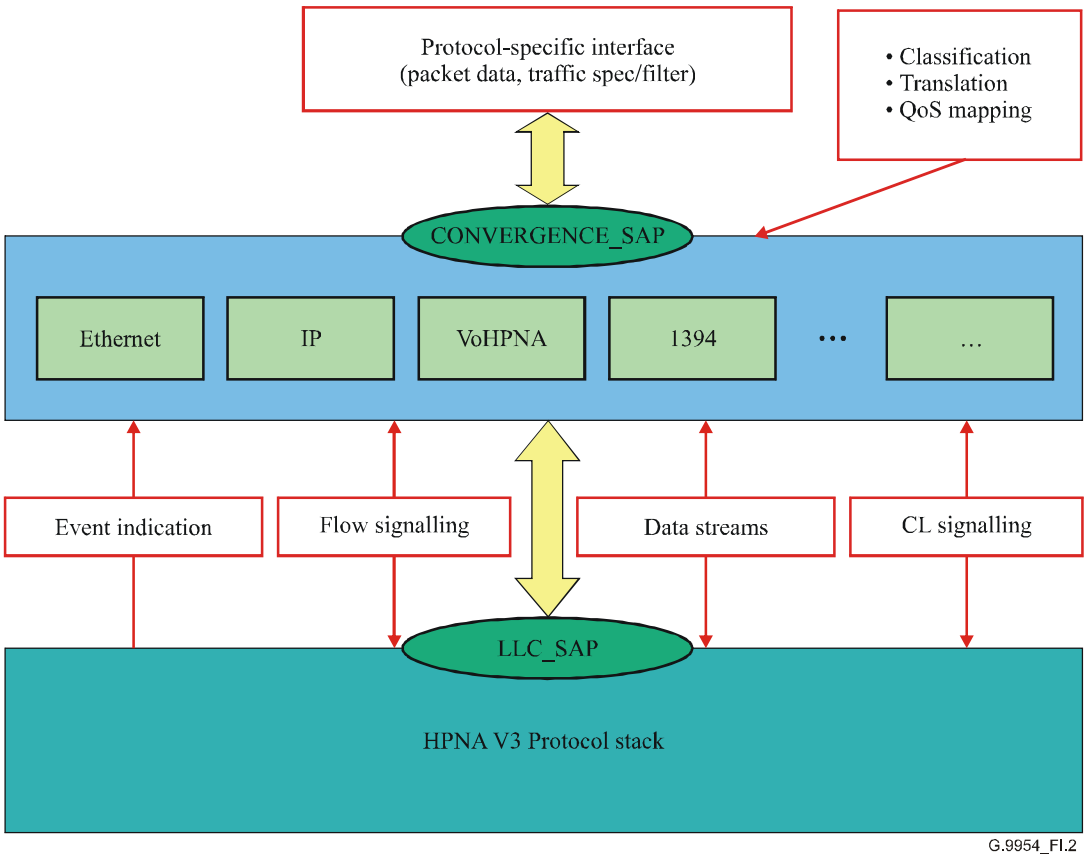


**Figure I.2/G.9954 – Convergence layer – Link-Layer primitives**

### I.2.1 Flow Signalling Primitives

#### I.2.1.1 LLC_SETUP_FLOW { req, cnf, ind, rsp }

This primitive is used to set up a flow between a source and a single or multiple destinations on the network. It is protocol specific as to which event at the protocol level will cause the setup of a flow and what are the flow characteristics.

The **request** primitive is used by the Convergence Layer to request the setup of a flow with defined flow properties and traffic classifier specification (see 9.2 above). If the source of the flow is also the device requesting the flow setup, the traffic classifier specification only has local significance. The **request** primitive is normally only generated at the source or destination of a flow although it is possible that it may be generated by the master.

The **indication** primitive is used to notify the Convergence Layer of the setup of a flow. The flow properties and traffic classifier are passed to the Convergence Layer. The flow properties delivered to the Convergence Layer are after admission control and contain the offered QoS properties and assigned *Flow ID*. The **indication** primitive may be used to trigger signalling operations with the higher-layer protocols and to initialize, install or populate protocol-specific data-structures such as address translation and bridging tables.

The **response** primitive is used by the Convergence Layer to signal to the Link Layer of the status of the flow setup request from the perspective of upper-layer protocols. It provides an opportunity for the upper-protocol layer to reject the flow setup request or offered flow properties due to some protocol-specific consideration.

The **confirm** primitive is used to notify the requestor of the status of the **request** and to report back information concerning the flow, including the *Flow ID* and offered flow parameters. The actual (offered) flow parameters may vary from the original request due to resource limitations.

The parameters in Table I.1 are used in this primitive:

**Table I.1/G.9954 – Primitive parameters**

| Parameter | Request | Confirm | Indication | Response |
|---|---|---|---|---|
| Flow Properties | √ | √ | √ | √ |
| Traffic Filter Specification | √ | | √ | |
| Status | | √ | | √ |

where:

- Flow properties – Properties of flow to be set up (see QoS spec). The Convergence Sub-layer participating in the interface is specified in the Flow Properties parameter as is the Flow ID assigned to the flow.
- Traffic filter specification – Filter specification as defined in QoS spec. Action specification for the filter is ADD.
- Status – Status of setup request in confirm primitive type.

For further information, see 10.17.

#### I.2.1.2 LLC_MODIFY_FLOW { req, cnf, ind, rsp }

The **request** primitive is used to request the modification of a flow's properties or the associated traffic classifier filters. The flow is identified by *Flow ID* in the flow properties parameter.

The **indication** primitive is used to notify the Convergence Layer of the requested modifications. Flow properties are after admission control. The traffic classifier filter specification may indicate an add, modify or delete action. This primitive may trigger operations within the upper layer protocol and may cause modifications of internal data structures.

The **response** primitive allows the Convergence Sub-Layer to accept or reject the modification request.

The **confirm** primitive is used to inform the Convergence Layer of the result of the request.

The parameters in Table I.2 are used in this primitive:

**Table I.2/G.9954 – Primitive parameters**

| Parameter | Request | Confirm | Indication | Response |
|---|---|---|---|---|
| Flow Properties | √ | √ | √ | √ |
| Traffic Filter Specification | √ | | √ | |
| Status | | √ | | √ |

where:

- Flow properties – Properties of flow to be modified (see QoS section). The Flow ID of the flow to be modified is encoded in the Flow properties.
- Traffic filter specification – Specification of the filter used to map to flow. Actions defined for filter specification includes add, modify and delete a filter.
- Status – Status of modify request in confirm primitive type.

For further information, see 10.17.3.2.

#### I.2.1.3 LLC_TEARDOWN_FLOW { req, cnf, ind, rsp }

This primitive is used to tear down an existing flow identified by *Flow ID*.

The parameters in Table I.3 are used in this primitive:

**Table I.3/G.9954 – Primitive parameters**

| Parameter | Request | Confirm | Indication | Response |
|---|---|---|---|---|
| Source MAC address | √ | √ | √ | √ |
| Destination MAC address | √ | √ | √ | √ |
| Flow ID | √ | √ | √ | √ |
| Status | | √ | | √ |

where:

- Source MAC address – Address of device at source of the flow.
- Destination MAC address – Address of device at destination of the flow.
- Flow – Identifies the flow to be torn down.
- Status – Status of modify request in confirm primitive type.

For further information, see 10.17.3.3.

### I.2.2 Data stream primitives

#### I.2.2.1 LLC_DATA { req, cnf, ind }

This primitive is used to send packet data between peer Convergence Sub-Layer entities.

The **request** primitive is used to request the transfer of a protocol layer packet or Convergence Layer information to a peer Convergence Layer entity over a particular flow (identified by *Flow ID)* or using a particular *priority* (if operating in master-less mode).

The **indication** primitive is used to notify the Convergence Sub-Layer of the arrival of the Convergence Layer information. The notification includes the timestamp at the time of reception measured with reference to a common point in the transmission frame. The point defined is immediately following the SA in the frame in which the frame arrived.

The **confirm** primitive is used to notify the completion of the data transfer request. Parameters of the primitive include the *Status* of the request and the timestamp when the data was actually transmitted on the media.

The parameters in Table I.4 are used in this primitive:

**Table I.4/G.9954 – Primitive parameters**

| Parameter | Request | Confirm | Indication | Response |
|---|---|---|---|---|
| FC | √ | | √ | |
| DA | √ | | | |
| SA | | | √ | |
| EtherType | √ | | √ | |
| MAC Aggregation | √ | | | |
| Payload Length | √ | | √ | |
| Payload | √ | | √ | |
| FCS | √ | | √ | |

**Table I.4/G.9954 – Primitive parameters**

| Parameter | Request | Confirm | Indication | Response |
|---|---|---|---|---|
| TX Timestamp | | √ | | |
| RX Timestamp | | | √ | |
| Status | | √ | √ | |

where:

- FC – is the Frame Control and includes the Frame-Type and Frame Sub-type, Priority/Flow ID and PE.
- DA – is the Destination Address of the SDU.
- SA – is the Source Address of the SDU.
- EtherType – is the Ethernet Type defined for the frame.
- MAC Aggregation – indicates whether the packet should be aggregated by the MAC-layer with other packets belonging to the same priority or flow. This parameter is used to indicate either no aggregation should be performed (a value of 0), or the packet is a candidate for aggregation (a value of 1).
- Payload – is the payload data to be delivered by the protocol stack. This payload may come from the Link Layer or Protocol Convergence Layers of the protocol stack. The payload frame format is not necessarily an Ethernet frame and may come from any convergence layer as indicated by the FT parameter.
- Payload Length – is the length of the payload data.
- FCS – is an optional 32-bit frame checksum that may be supplied with the frame.
- TX Timestamp – Timestamp of actual transmission. Time is specified in units of $2^{-13}$ ms.
- RX Timestamp – Timestamp of actual reception. Time is specified in units of $2^{-13}$ ms.
- Status – is the data TX/RX status.

### I.2.3 Event indication primitives

#### I.2.3.1 LLC_MAC_CYCLE { ind }

This primitive is used to notify the Convergence Layer of MAC cycle timing information and of Media Access Planning (bandwidth allocations). The primitive provides information that enables Convergence Layers to synchronize upper-protocol layers with the G.9954 MAC cycle, synchronize sampling rates and use media resource allocation information for protocol level signalling.

This primitive is intended for use in Convergence Layers that interface with upper-layer protocols that are synchronous in nature or support isochronous services and require some degree of synchronization. Examples of such protocols include IEEE 1394, USB, etc.

The parameters in Table I.5 are used in this primitive:

**Table I.5/G.9954 – Primitive parameters**

| Parameter | Request | Confirm | Indication | Response |
|---|---|---|---|---|
| MAP | | | √ | |
| Scheduled MAC Cycle Start Time | | | √ | |
| Actual MAC Cycle Start Time | | | √ | |
| Indication Time | | | √ | |

where:

- MAP – is the MAP control frame.
- Scheduled MAC Cycle Start Time – is the time when the MAC cycle was scheduled to start.
- Actual MAC Cycle Start Time – is the time when the MAC cycle actually started. This may differ from Scheduled MAC Cycle Start Time if jitter was introduced into the MAC cycle due to AMAC interference.
- Indication Time – is the time when the indication was actually delivered to the Convergence Layer.

For a further description of the parameters used in the LLC_MAC_CYCLE primitive, see the description of the MAP in 10.14.1.

#### I.2.3.2 LLC_NETWORK_ENTRY { ind }

This primitive is used to notify the Convergence Layer of the registration of the device with the master and of the assigned Device ID.

The parameters in Table I.6 are used in this primitive:

**Table I.6/G.9954 – Primitive parameters**

| Parameter | Request | Confirm | Indication | Response |
|---|---|---|---|---|
| Device ID | | | √ | √ |
| 802.3 MAC address | | | √ | √ |

where:

- Device ID – is the master-assigned device ID.
- 802.3 MAC address – is the 48-bit IEEE MAC address assigned to the node.

#### I.2.3.3 LLC_NETWORK_EXIT { ind }

This primitive is used to notify the Convergence Layer of the de-registration of a device with the master

The parameters in Table I.7 are used in this primitive:

**Table I.7/G.9954 – Primitive parameters**

| Parameter | Request | Confirm | Indication | Response |
|---|---|---|---|---|
| Device ID | | | √ | |

where:

- Device ID – is the master-assigned Device ID.

#### I.2.3.4 LLC_SYNC_EVENT { ind }

This primitive is used to notify the Convergence Layer of the synchronization of a G.9954 device with a master-generated MAC cycle.

**Table I.8/G.9954 – Primitive parameters**

| Parameter | Request | Confirm | Indication | Response |
|---|---|---|---|---|
| Sync Event | | | √ | |

### I.2.3.5 LLC_SYNC_LOSS_EVENT { ind }

This primitive is used to notify the Convergence Layer of the loss of synchronization with the master-generated MAC cycle.

**Table I.9/G.9954 – Primitive parameters**

| Parameter | Request | Confirm | Indication | Response |
|---|---|---|---|---|
| Sync Loss Event | | | √ | |

## I.3 Convergence layer architecture

The internal structure of the Convergence Layer component according to the model described above is illustrated in Figure I.3:

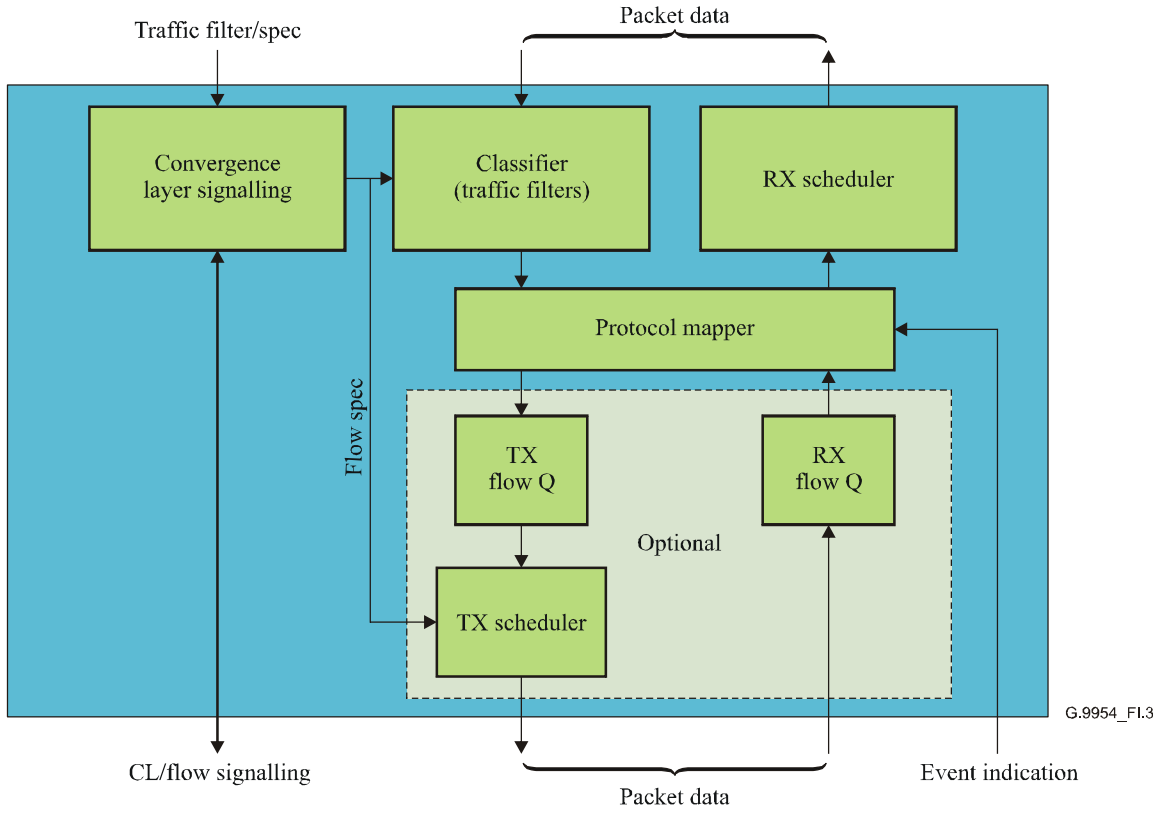


**Figure I.3/G.9954 – Convergence Layer architecture**

The components within the Convergence Layer block are responsible for the following functions:

- **Flow/Convergence Layer Signalling** – This component is responsible for performing flow setup/teardown signalling and peer convergence-sublayer signalling. It responds to flow setup requests, originating from upper-protocol layers or from within the Convergence Layer itself, and manages Convergence Layer peer-level signalling. It communicates with the Classifier in order to define traffic-filter specifications and with the TX scheduler in order to define traffic-rate specification.
- **Classifier** – The Classifier is responsible for mapping incoming packet data to a flow using the traffic filter specification defined by the Convergence Layer Signalling component.
- **Protocol Mapper** – This component is an optional entity and may perform protocol-specific mapping functions.

- **Flow Queues** – Flow queues are optional data-structures used to hold packets while they are waiting to be scheduled by the appropriate scheduler component. Flow queues on the TX side may be token buckets that are used for traffic shaping.
- **TX Scheduler** – The TX scheduler is responsible for selecting packets from the TX flow queue and delivering it to the underlying network device. It may perform traffic shaping functions. This function may be trivial in those implementations that do not require flow queues and shaping in the Convergence Layer.
- **RX Scheduler** – The RX scheduler is responsible for delivering packets received from the network interface to the upper-protocol layers. Packets arriving from the network may be passed through the Protocol Mapper in order to perform the inverse protocol mapping function.

Convergence sub-layers may need to maintain several data-structures in order to implement sub-layer functions. Examples of such data-structures include traffic queues, used to shape traffic according to rate parameters, buffers used to balance the differences in cycle frequencies between upper and lower level protocols, address mapping tables used for bridging between networks, etc.

Since memory demands for certain Convergence Sub-layers may be significant, it is possible that Convergence Sub-layers may be implemented at the host driver level and not on-chip.

### I.4 Flow setup triggering

Flow setup may be triggered by the following events:

- Registration of device with master;
- Arrival of an upper-layer Service Data Unit (SDU);
- Upon request from upper protocol layer;
- Management operations.

In the first case, when flow setup is triggered by the registration process, the operation may be initiated in either the master or the endpoints. In both cases, the assumption is that the master and/or endpoint knows which flows need to be provisioned after registration and what are the flows properties. This information may be built-in to the Convergence Layer or it may be attained from configuration parameters.

In the second case, when a flow is set up upon arrival of an SDU, the assumption is that the Convergence Layer has traffic filters installed that allow it to classify an SDU upon arrival and to identify the properties of the flow that needs to be set up to handle traffic of this type. It then should initiate the flow setup using the flow specification attached to the filter. The filters and their association with the flow property descriptor may be built into the Convergence Layer or it may be installed in a configuration data.

Upper layer protocols may also initiate the setup of a flow with specific properties. For example, applications may initiate flow setup in response to handling RSVP or equivalent DOCSIS signalling messages.

Management operations, whether initiated from the local or remote sides of the device, may initiate the set up of a flow with well-defined flow properties.

### I.5 Classification

Classification is the process by which upper-layer PDUs are mapped to G.9954 flows. The classification process is protocol-specific and may include a set of classification rules that are processed in a particular priority ordering.

The classification rules that apply to a flow are part of the flow description. This model is consistent with the RSVP model that defines a *flow descriptor* as the composite of a *flow specification* (the traffic-related component) and a *filter specification.*

For a description of traffic classification filters, see 9.3 above.

## I.6 Convergence Layer interfaces to upper-protocol layers

Each convergence sub-layer provides its own protocol specific interface to the upper layer. All interfaces provide a primitive (or primitives) for transporting and receiving the upper-layer Protocol Data Units. The primitives in this interface are of the form:

- XXX_CSL_DATA.req – Used to request the transmission of data.
- XXX_CSL_DATA.cnf – Used to notify the upper layer of the status of the transmission request.
- XXX_CSL_DATA.ind – Used to notify the upper layer XXX of the arrival of data.

## I.7 Protocol-Specific convergence layers

### I.7.1 IP convergence

The IP convergence layer processing may use RSVP protocol packet filtering rules. These rules specify classification according to the following criteria:

- IP Type-Of-Service (TOS) field;
- IP Protocol Number;
- IP Source Address;
- IP Destination Address;
- IP Protocol Source Port Number;
- IP Protocol Destination Port Number.

For further details on IP traffic classifiers see 10.17.

### I.7.2 Ethernet convergence

Ethernet convergence layer processing performs classification of PDUs based on the following criteria:

- Ethernet Destination MAC Address;
- Ethernet Source MAC Address;
- Ethernet Type and 802.2 SAP;
- VLAN (802.1P) priority;
- VLAN (802.1Q) ID.

The special Ethernet Types in Table I.10 are recognized by the Ethernet convergence layer and result in PDUs being routed to the appropriate convergence layer component:

**Table I.10/G.9954 – Routed Ethernet Types**

| Ethernet Type | Description |
|---|---|
| 0x0800 | IP packet routed to IP convergence layer |
| 0x0806 | ARP packet routed to IP convergence layer |
| 0x86DD | Ipv6 packet routed to Ipv6 convergence layer |

For further details on Ethernet traffic classification filters, see 10.17.

### I.7.3 IEEE 1394 (Firewire) convergence

The primitives for this Convergence Sub-Layer are for further study.

### I.7.4 Universal Serial Bus (USB) convergence

The primitives for this Convergence Sub-Layer are for further study.

# Appendix II

# Media Independent Interface (MII) Recommendations

The Media Independent Interface (MII) as specified in IEEE Std 802.3-1998, clause 22, is a common interface found on many pieces of existing networking silicon. While there are many possible implementations for interfacing an G.9951/2 PHY to an existing Ethernet MAC via the MII, the following guidelines provide a reference for designing a PHY that is completely compatible with silicon complying with clause 22.

Flow control is the major issue in using the MII interface. The MII specification calls for interface clocks to be fixed frequency of 25 MHz ± 100 ppm, resulting in a data transfer rate of 100 Mbit/s. This Recommendation provides for a wide range of bit rates ranging from 4 Mbit/s to 128 Mbit/s. For the PHY-to-MAC (receive) direction there is a rate mismatch between the PHY and MAC over this interface. This may result in some packet loss in the unlikely event that transmissions on the wire are all at full rate. In this case a receiver should limit the maximum size of its frame receive buffer in order to force transmitters to transmit shorter frames and to guarantee that the effective throughput does not exceed the MII 100-Mbit/s limit. For the MAC-to-PHY (transmit) direction, the PHY needs a method to hold off the MAC while previous data is being modulated and sent out on the wire.

This flow control should use the CRS signal in a "false carrier sense" mode to hold off the MAC transmitter with the Deference mechanism. The details of this signalling are described below.

## II.1 MII overview

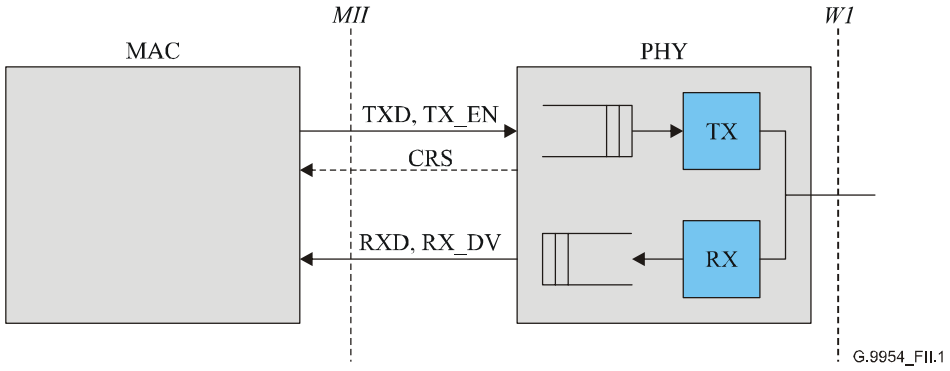


**Figure II.1/G.9954 – MII interface**

### II.1.1 MII data path

The MAC/PHY interface consists of the following 16 signals:

**Table II.1/G.9954 – MAC/PHY signals**

| Signal | Direction relative to PHY | Description |
|---|---|---|
| TX_EN | In | Transmit framing signal |
| TXD[3:0] | In | Four bits per clock of transmit data |
| TX_ER | In | Transmit error |
| TX_CLK | Out | Transmit clock (2.5 MHz or 25 MHz) |
| CRS | Out | Carrier sense |
| RX_DV | Out | Receive data valid |
| RXD[3:0] | Out | Four bits per clock of receive data |
| RX_CLK | Out | Receive clock |
| RX_ER | Out | Receive error |
| COL | Out | Collision |

### II.1.2 Transmission without collision

Shown in Figure II.2 is an example transfer of a packet from MAC to PHY.

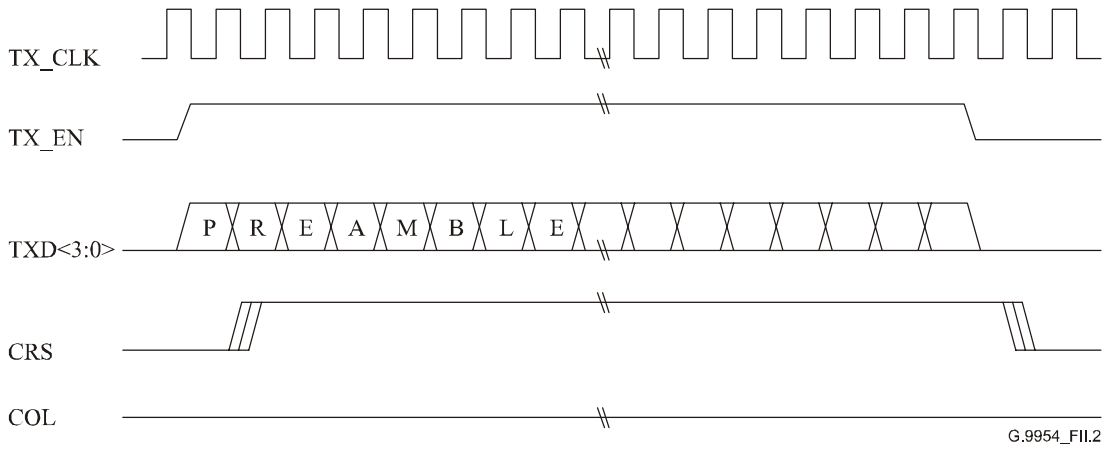


**Figure II.2/G.9954 – MAC to PHY packet transfer**

### II.1.3 Reception without error

Shown in Figure II.3 is an example of transfer of a packet from PHY to MAC.

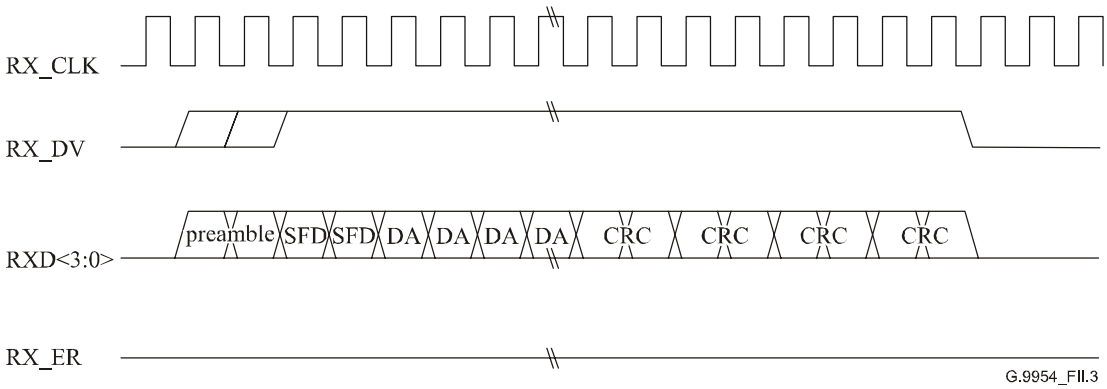


**Figure II.3/G.9954 – PHY to MAC packet transfer**

### II.1.4 MII management signals

There are two additional signals specified for management: MDIO (Management Data I/O) and MDC (Management Data Clock). Many, if not all, existing MACs will have MDIO/MDC interface pins, but these are vital only for management purposes in the case there are registers in the PHY that need to be accessed by the host. There is generally no requirement for MII-based management functions and, consequently, it is not necessary for PNT devices to implement MDIO/MDC. However, if there are registers in the PHY, then the protocol and signalling defined by MDIO/MDC in IEEE Std 802.3 clause 22, should be used.

## II.2 G.9951/2 signalling Recommendations

The following description references clause 22, Media Independent Interface specification, used in the 100 Mbit/s half-duplex mode. To account for physical layer differences between G.9951/2 and 100BASE-T Ethernet, the PHY is assumed to have an adaptation or reconciliation layer which handles all timing and data formatting issues. The MII is used as a data channel that transfers data back and forth in units of packets, flow controlled by the carrier sense (CRS) signal.

### II.2.1 TX_CLK and RX_CLK

The PHY generates a stable, continuous 25 MHz square wave which is supplied to TX_CLK and RX_CLK. No "gapping" or other variable clocking method is used.

The frequency offset of the generated clock should be controlled to enable the use of all standard MAC implementations.

### II.2.2 TX_ER and RX_ER

TX_ER is normally used in situations where the transmitter above the PHY has detected an error condition, but the transmission is currently in process. TX_ER indicates to a PHY that the current packet is errored and should be corrupted on the wire to ensure a receiver does not accept this as a valid packet. Normally, this is a condition that only applies to repeaters. Repeaters do not perform error checking on the complete packet. In the case of a DTE (sometimes referred to as a 'node'), the transmitter usually guarantees the frame to be without errors and there is no need for the TX_ER signal. Since G.9951/2 is based on bus topology wiring plants, no repeater is specified and use of the TX_ER signal is not anticipated. However, G.9951/2 PHYs may choose to respond to the TX_ER signal.

RX_ER is normally used in situations where the PHY detects an error in the receive stream as a result of decoding. G.9951/2 PHYs may assert this signal in the event that such an error is detected.

### II.2.3 TX_EN

TX_EN from the MAC provides the framing for the Ethernet packet. TX_EN active indicates to the PHY that data on TXD[3:0] should be sampled using TX_CLK.

### II.2.4 TXD[3:0]

TXD[3:0] contains the data to be transmitted and transitions synchronously with respect to TX_CLK. TXD[0] is the least significant bit. It is generally assumed that the data will contain a properly formatted Ethernet frame. That is, the first bits on TXD[3:0] correspond to the preamble, followed by SFD and the rest of the Ethernet frame (DA, SA, length/type, data, CRC).

The PHY strips the 802.3 preamble on MAC-to-PHY transfers.

### II.2.5 RX_DV

RX_DV is asserted by the PHY to indicate that the PHY has decoded receive data to present to the MAC.

### II.2.6 RXD[3:0]

RXD[3:0] contains the data recovered from the medium by the PHY and transitions synchronously with respect to RX_CLK; RXD[0] is the least-significant bit. It is assumed that the PHY has properly formatted the frame such that the MAC will be presented with expected preamble plus SFD.

The TXD and RXD data paths are full duplex, although we use the MII interface in half-duplex mode. RX_DV is never asserted at the same time as TX_EN.

### II.2.7 CRS

On transmit, the PHY asserts CRS some time after TX_EN comes true, and drops CRS after TX_EN becomes false AND when the PHY is ready to receive another packet. When CRS falls, the MAC times out an IFG (.96 microsecond) and may assert TX_EN again if there is another packet to send.

This differs from nominal behaviour of CRS in that CRS can extend past the end of the packet by an arbitrary amount of time, while the PHY is gaining access to the channel and transmitting the packet. See Figure II.4.

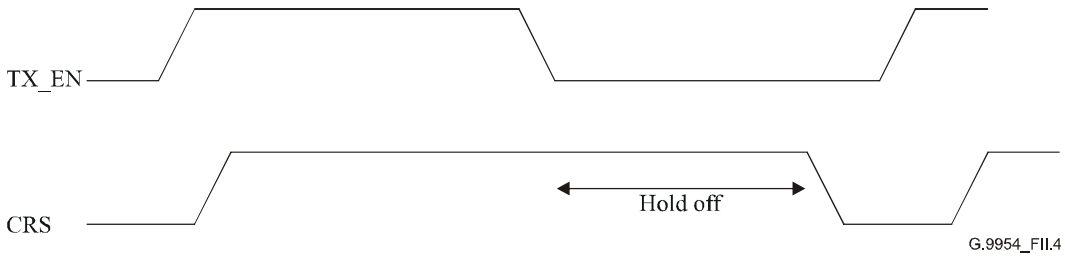


**Figure II.4/G.9954 – TX direction**

MACs in 100 Mbit/s mode do not use a jabber timeout, so there is no timing restriction on how long CRS can be asserted (other than sanity timeouts the PHY may implement).

Transmissions can "cut through" or begin to be modulated onto the wire as soon as the transfer begins, as the MII will fill the PHY buffer faster than data needs to be made available to the modulator. When a packet arrives at the PHY, it attempts to gain access to the channel using the priority CSMA/CD algorithm described in clause 7. This may not happen before the entire packet is

transferred across the MII interface, so the PHY will need to buffer at least one MTU to perform this rate adaptation.

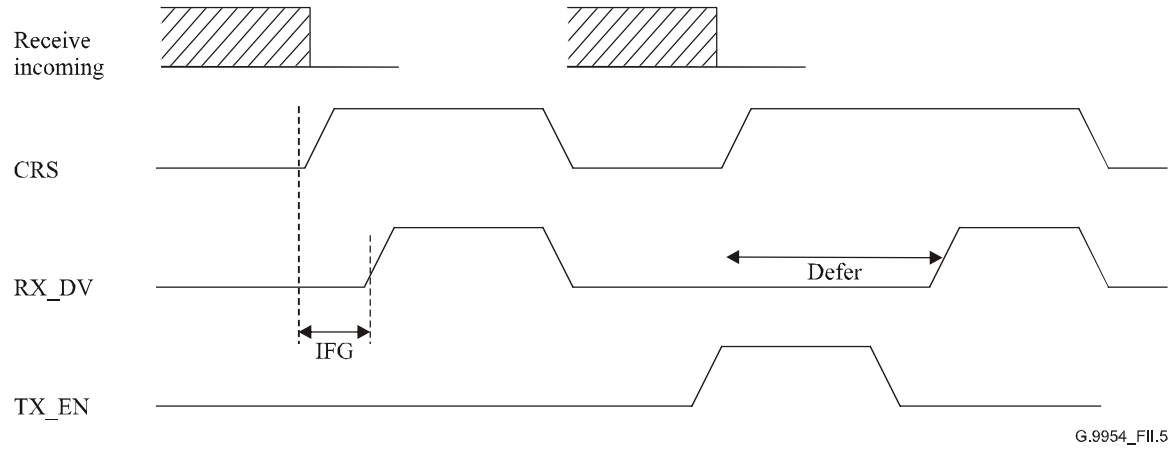


**Figure II.5/G.9954 – RX direction**

On receive, when the PHY anticipates that it will have a packet demodulated it raises CRS to seize the half-duplex MII channel, waits a short time (an IFG), then possibly defers to TX_EN (which may just have been asserted) plus an IFG, and then raises RX_DV to transfer the packet. At the end of the transfer, it drops CRS unless the transmit buffer is full or there is another receive packet ready to transfer. (See Figure II.5, where one receive transfer is followed by a second which defers to TX_EN.)

RX_DV should not be asserted until the PHY is assured that the entire packet will be ready to transfer at the 100 Mbit/s rate. This implies some buffering on the receive side to do this rate adaptation. Once the MII burst transfer starts, new data can start filling the buffer, as the MII transfer is guaranteed to stay ahead of the data coming off the wire.

Receive direction transfers need to have priority over transmit direction to ensure that the buffer empties faster than packets arrive off the wire. The longest that the receiver needs to wait is the time to transfer one TX frame plus an IFG or approximately 134 microseconds. However, minimum size frames can arrive at a peak rate of one every 65 microseconds, so the receive side buffer has to accommodate multiple frames (but only little more than one MTU of data).

### II.2.8 COL

COL is not used. The way the PHY manages the MII interface, collisions between receive and transmit direction transfers do not occur.

## II.3 The "Off-Chip" G.9954 convergence layer

Interfaces to external MACs in G.9954 relies on the implementation of protocol-specific convergence layers. The separation of the convergence layer from the G.9954 Link and MAC sub-layer facilitates the tailoring of external protocols and interface implementations to G.9954. Furthermore, it lends itself to an off-chip solution where the convergence layer logic resides in the host driver software. In such an environment, where memory requirements may be more relaxed, the convergence layer may be used to hide the complexity of the interface between an external MACs and the G.9954 device.

The following clauses describe the "off-chip" convergence layer architecture and how it can be transparently embedded in a software driver environment based on the NDIS or similar architecture. A discussion of MII interface implementation issues follows.

In configurations, where the complexity of the interface is limited or standard software drivers are used, convergence layer functions should be performed in an "intermediate software driver", running in the host operating system at a level between the host's "standard software driver" and the hardware interface. In such a configuration, the "intermediate software driver" should be made responsible for performing packet buffering and "traffic-shaping" in order to guarantee that packets are delivered to the hardware at a data-rate that does not exceed the traffic specification of the active flows.

The architectural model that has the G.9954 convergence layer running off-chip in the "intermediate software driver" is described in Figure II.6:

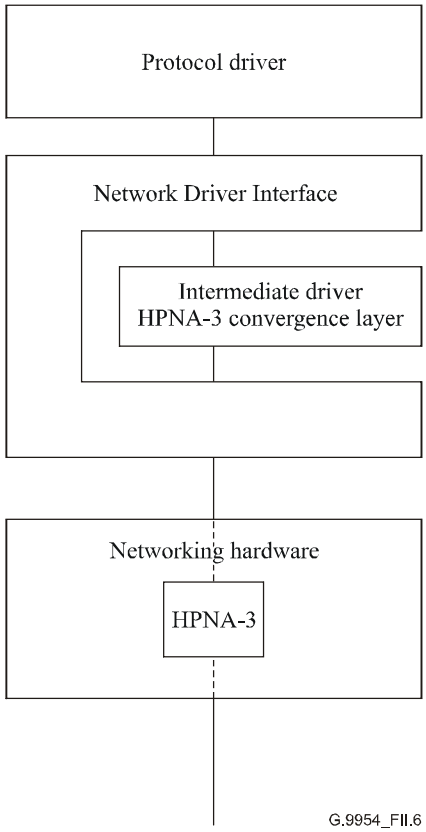


**Figure II.6/G.9954 – "Off-chip" convergence layer**

This model assumes the existence of a Network Driver Interface that is located between the protocol driver (e.g., 802.3 driver) and the actual networking hardware. It also assumes that there is a way to interface the Intermediate Software Driver into the Network Driver Interface in a transparent manner such that all packets that reach the Network Driver Interface from the Protocol Driver or Networking Hardware are diverted through the Intermediate Driver.

The Intermediate Driver model is convenient for performing the following kinds of functions:

- Protocol Translation – Map packets between protocol formats. May include bridging and address translation tables, etc.
- Packet Filtering – A traffic shaper and/or scheduler may be used to buffer incoming packets and reorder their delivery to the underlying networking hardware.

Using this model, intelligence in the Intermediate Software Driver allows the underlying interface to the G.9954 chip to be simple and standard, such as one based on the MII interface. Packets delivered to the MII interface can be safely blocked if no more memory resources exist since traffic

shaping algorithms guarantee that data will not be delivered at a rate that is greater than the negotiated rate of the active flows.

The NDIS (Network Driver Interface Specification) driver model conforms to the above architecture.

# Appendix III

# End-to-end architecture

## III.1 G.9954 to G.9954 protocol stack

Figure III.1 shows an end-to-end protocol stack involving two interconnected G.9954 devices. Each G.9954 device has a 48-bit MAC address. Each protocol layer exchanges protocol messages over a virtual link with PNT PHY's being connected physically over a phone-wire or cable media network.

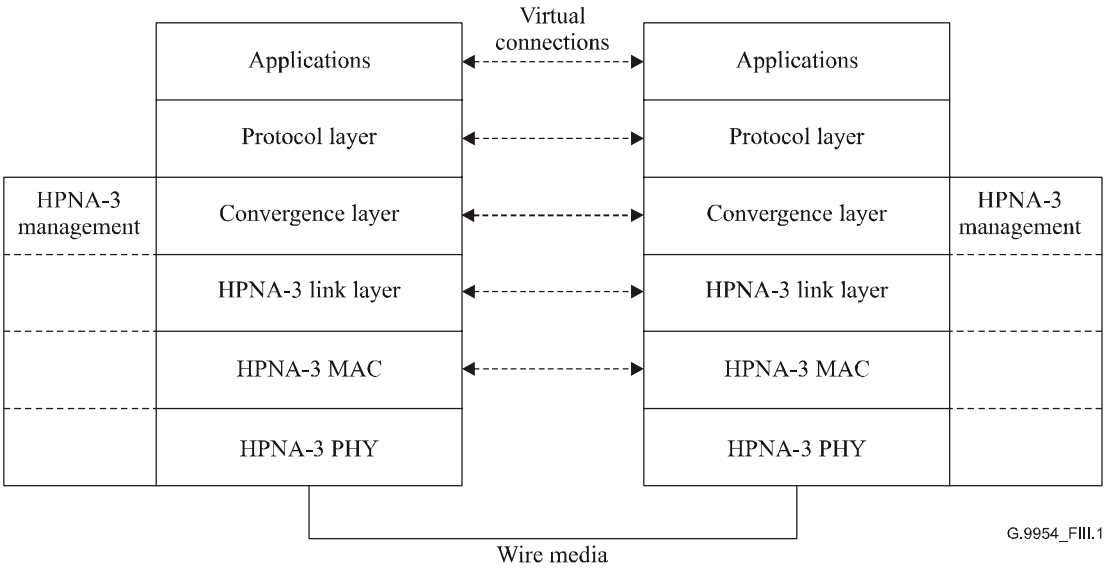


**Figure III.1/G.9954 – Communicating G.9954 protocol stacks**

## III.2 Ethernet-PNT interface

Ethernet is the natural protocol for transport over a PNT network. The PNT frame format is an extension to the Ethernet frame format and includes the entire Ethernet PDU within the frame.

G.9954 may interface with the Ethernet protocol in the following configurations:

- Ethernet PHY (MII Interface);
- Ethernet-PNT Bridge (MII Interface);
- Integrated Ethernet MAC-PHY (NIC card PCI or similar).

In the first configuration, G.9954 presents an MII interface and masquerades as an Ethernet PHY. This supports a glue-less connection to an external Ethernet MAC chip as illustrated in Figure III.2.

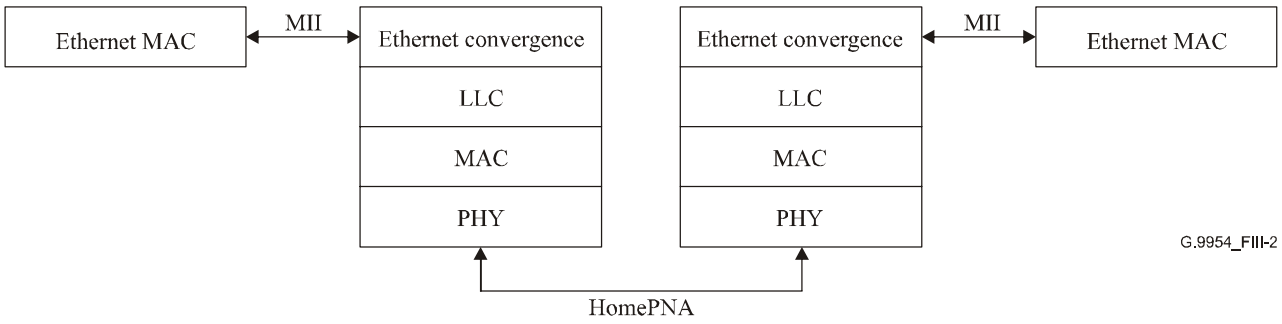


**Figure III.2/G.9954 – Ethernet PHY emulation**

In another configuration, G.9954 provides an MII interface to an on-chip Ethernet MAC bridge. This interface is suitable for connecting to an Ethernet PHY in order to build an Ethernet-PNT bridge, as shown in Figure III.3:

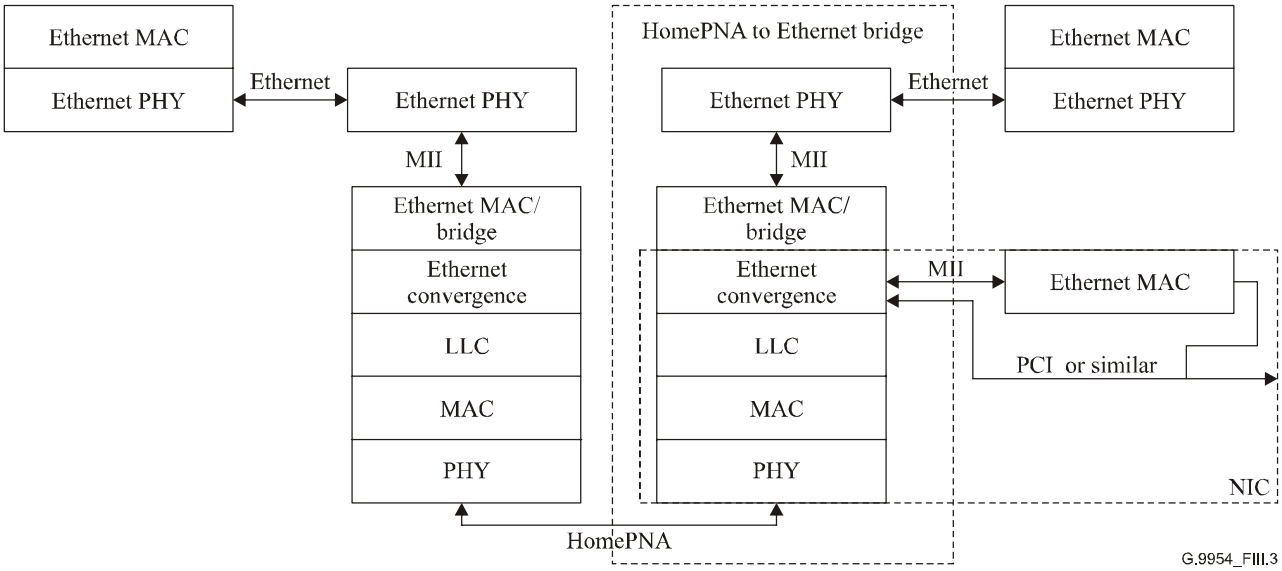


**Figure III.3/G.9954 – Ethernet-PNT bridge and NIC applications**

## III.3 USB to G.9954 protocol stack

A USB to G.9954 Adapter (dongle) is a USB device that provides a G.9954 connection to the host system. In this sense, it provides the same capability as a Network Interface Card (NIC) except that the Host PC connects to the network using the USB serial bus rather than the PCI bus.

USB is different from network protocols, such as Ethernet or IEEE 1394 in the sense that it is not an end-to-end network protocol but rather a bus protocol used to transfer data and control information from a host to the USB device. Data transfers, once they have arrived at the USB device, are removed from their USB wrappers, reconstructed into packets and transported over the PNT network. The USB wrappers themselves are discarded at the USB device endpoint.

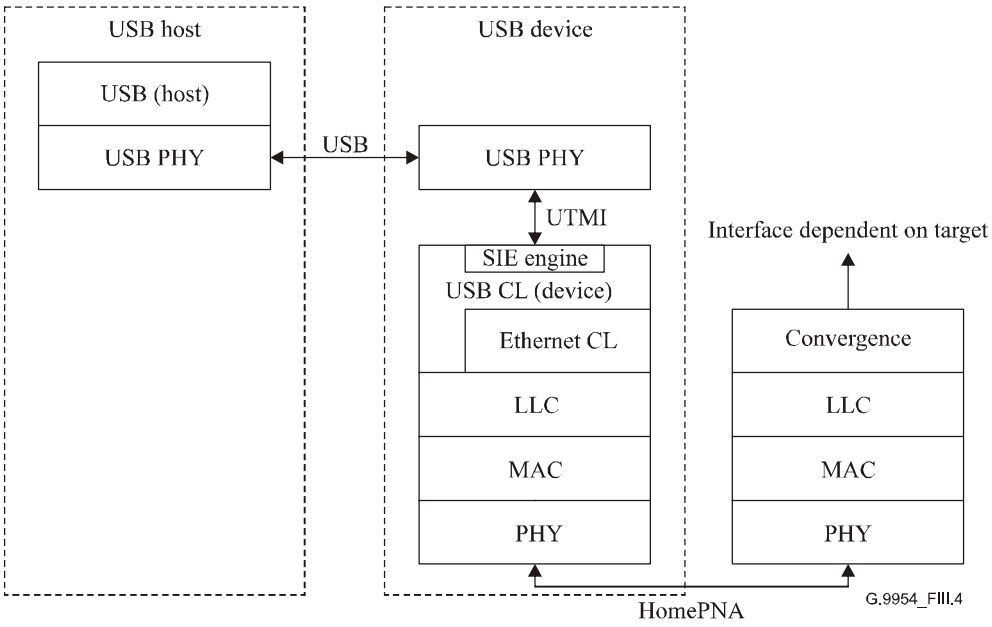


**Figure III.4/G.9954 – USB to G.9954 protocol adapter**

## III.4 IEEE 1394 to G.9954 protocol stack

Two architectures incorporating IEEE 1394 and G.9954 are considered:

- IEEE 1394 over G.9954;
- IEEE 1394-G.9954 bridge.

In the first architecture, the G.9954 device presents an IEEE 1394 Link Layer Interface to the IEEE 1394 protocol stack allowing IEEE 1394 applications to run over G.9954 in a transparent manner, as if they were running over an actual IEEE 1394 Link and PHY layer. This implies that the 1394 Convergence Layer implements the standard IEEE 1394 Link-Layer primitives and maps these primitives to G.9954 Link-Layer functions. This is illustrated in Figure III.5.

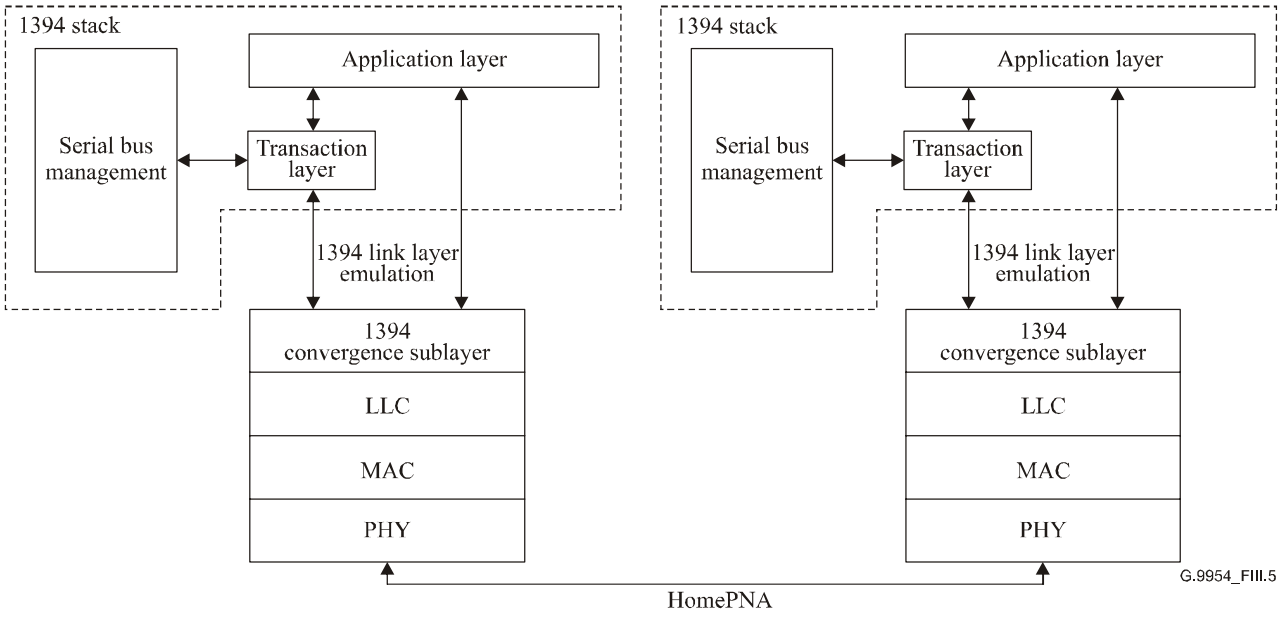


**Figure III.5/G.9954 – Transparent IEEE 1394 over G.9954**

The second architecture is used to interconnect an IEEE 1394 bus with the G.9954 network using the P.1394.1 standard (see [6]). In this configuration, the G.9954 convergence layer includes IEEE 1394 bridging functions for asynchronous and isochronous data in addition to the IEEE 1394 convergence layer described above. This is illustrated in Figure III.6.

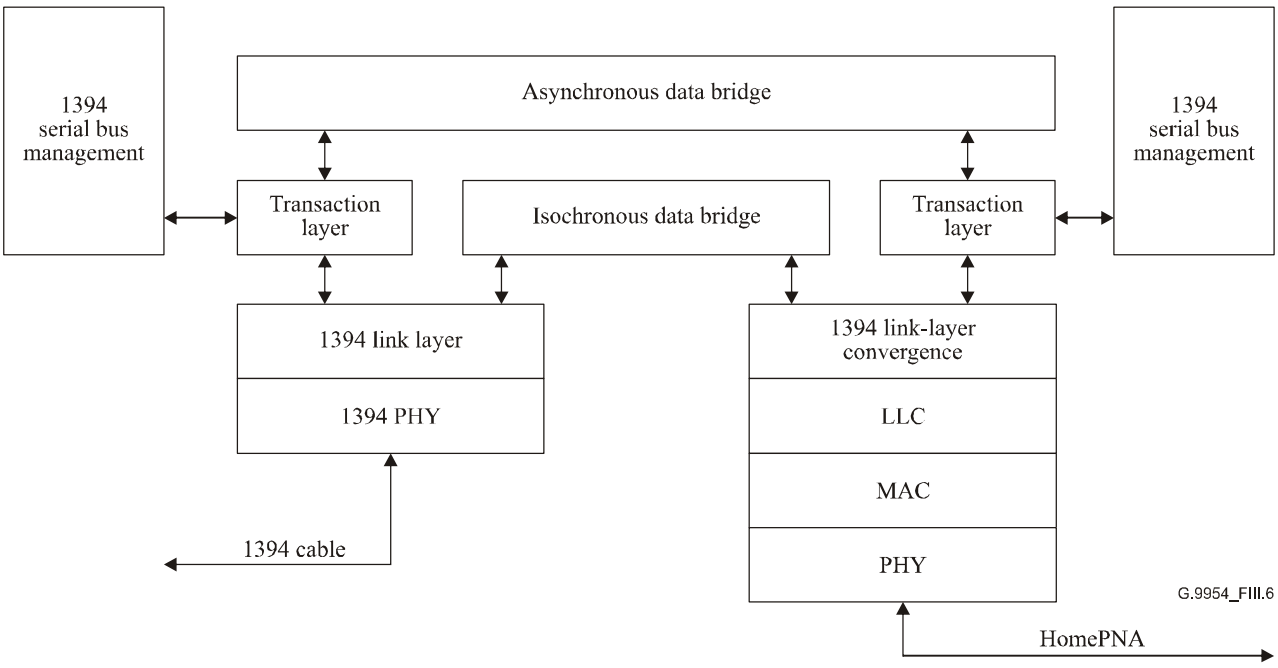


**Figure III.6/G.9954 – IEEE 1394-G.9954 bridge**

The details of this protocol bridge are for further study.

## III.5 DOCSIS to G.9954 protocol stack

The protocol stack for a DOCSIS to G.9954 Bridge described below is based on the DOCSIS specification for CPE-Controlled Cable Modems defined in [4] and the DOCSIS Radio Frequency Interface Specification in [5].

The first specification assumes a Cable Modem device connected to a Customer Premises Equipment (CPE) over an 802.3/Ethernet, USB or PCI PHY, which is used to transparently transport 802.3 MAC frames between Cable Modem and CPE devices. Since DOCSIS is defined as a system for the transparent transport of IP traffic over cable, the interface assumes that bridging between DOCSIS and other protocols, such as G.9954, is performed at the Ethernet/802.3 MAC frame level. This is illustrated in Figure III.7.

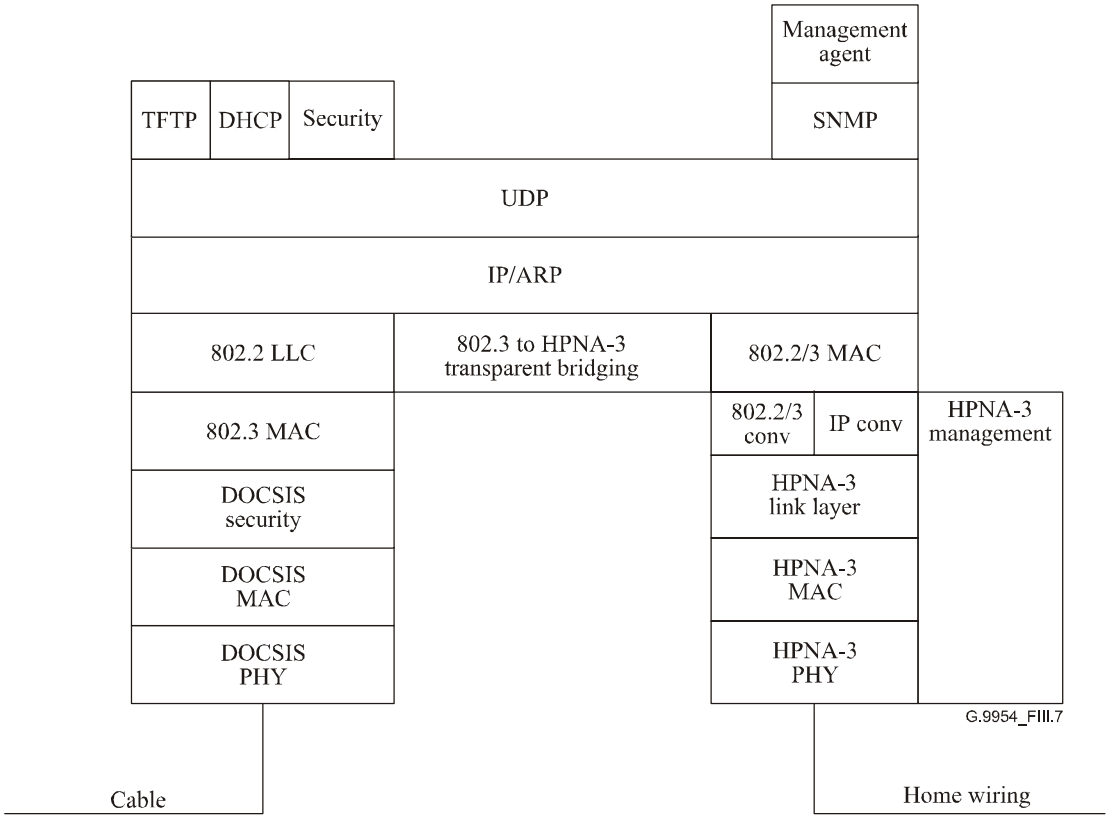


**Figure III.7/G.9954 – DOCSIS to G.9954 protocol stack**

An additional configuration involves a direct interface to the DOCSIS MAC. This is a lower-level interface than the Ethernet/802.3 interface and provides access to elements in the MAC Data Service Interface of DOCSIS, such as master CLOCK SYNCHRONIZATION, UPSTREAM GRANT SYNCHRONIZATION, that can be used to synchronize the G.9954 home network to the external DOCSIS network. This is illustrated in Figure III.8.

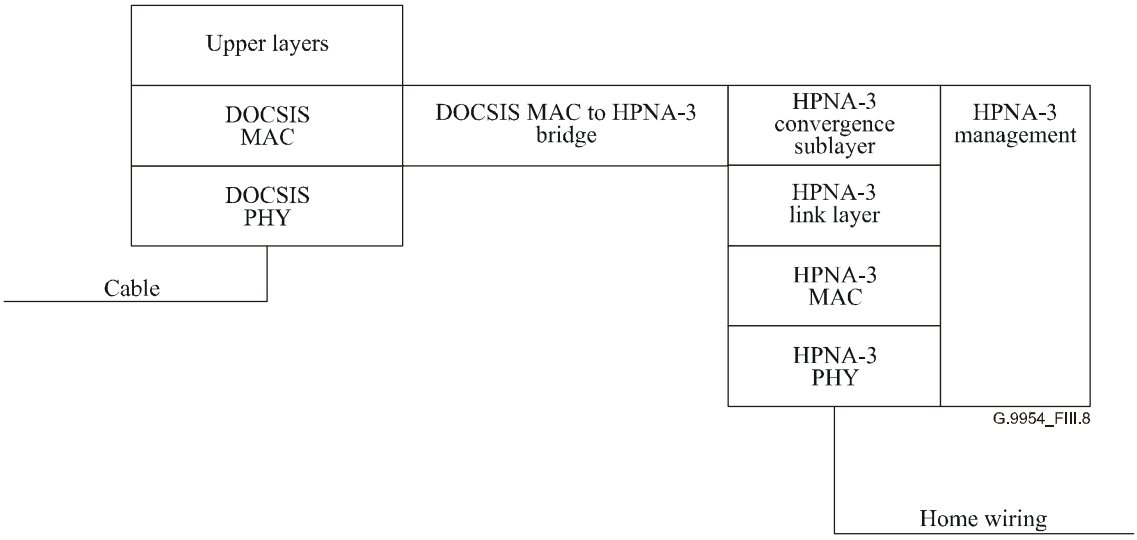


**Figure III.8/G.9954 – DOCSIS to G.9954 bridge**

# Appendix IV

# Network synchronization

The requirement to support synchronization to external network and protocols is derived from the types of services being delivered to the home and the networking technology and protocols used to transport these services. Given that some of these services, such as voice, audio and video are isochronous in nature and sensitive to latencies and jitter introduced by connecting networks, as well as to the differences in the clock frequencies between source and destination elements, in order to preserve the quality of a delivered service and to extend it into the home, a home networking technology should provide capabilities to allow synchronization of the home and external networks.

The proposed G.9954 protocol supports several built-in mechanisms that, when used together, support end-to-end synchronization of home networks with an external synchronous network and services. These mechanisms and the manner in which they interoperate are described in the following clauses.

## IV.1 Synchronization requirements

In order to synchronize elements connected to the home network to an external source or service, the following requirements must be addressed:

- Synchronization of data sampling rates – The frequencies of the clock used for data sampling at the source and destination of a service must be synchronized so as to guard against data underrun and overrun.
- Clock reference synchronization – Synchronization of clocks to common time reference may be required in order to relate to timestamp references that appear in sampled data or in protocol management messages.
- Synchronization to allocated timeslots and bandwidth grants – In order to reduce latency and jitter introduced by the home network, it is necessary to synchronize the allocation of timeslots on the home network with those on the external network used to deliver the service. The synchronization requirement is that data delivered to one network should only need to wait a minimal amount of time before gaining access to the other network.
- Quality of Service – Quality of Service mechanisms in the home network are required in order to guarantee timely access to the home network in accordance with QoS constraints of the delivered service.
- Protocol-awareness – In order to synchronize with external protocols it is necessary to have protocol-specific knowledge of the elements used for synchronization. For example, knowledge of clock synchronization services in IEEE 1394 or time-synchronization and timeslot grant information in DOCSIS.

## IV.2 The network synchronization model

The mechanisms used to support end-to-end synchronization to an external network are illustrated Figure IV.1.

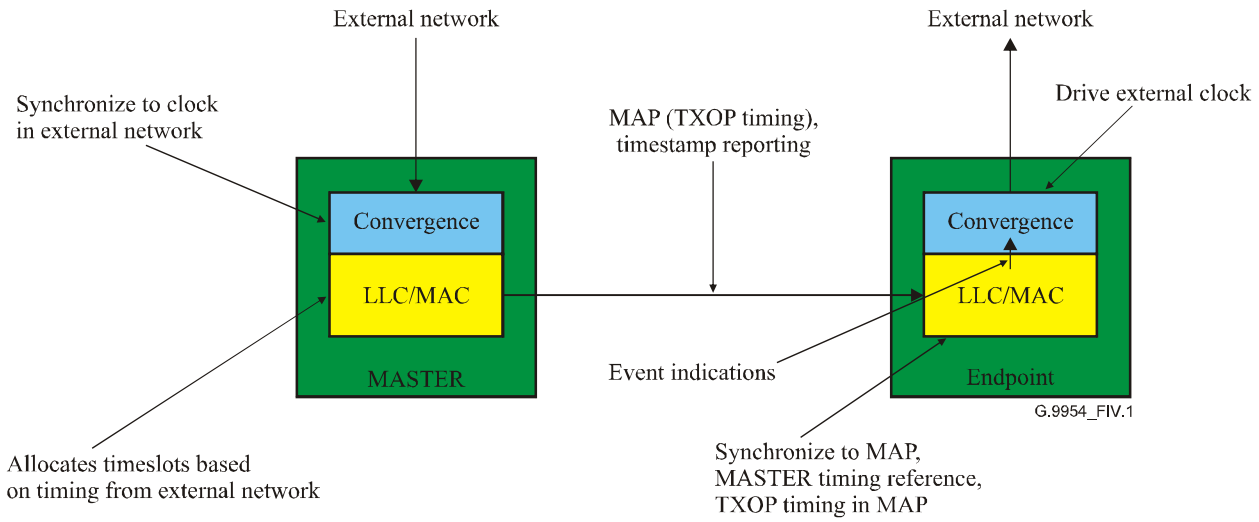


**Figure IV.1/G.9954 – Network synchronization model**

The model describes a network based on a master connected to an external network that delivers synchronous services, such as telephony or video services, and one or more endpoint (SLAVE) devices connected to the master on the home network.

In this model, the convergence layer on the master side has protocol-specific knowledge of the connected external network and uses this knowledge to derive a clock reference from the external protocol. This may involve processing of protocol-specific messages, such as DOCSIS Time Synchronization (SYNC) messages or accessing protocol-specific registers that implement clock synchronization services, as defined in IEEE 1394. This timing information can be used to "drive" the G.9954 system clock and synchronize its time-reference to that of the external network. This allows time-references derived from the external network, such as timestamp information, to be easily interpreted within the context of the home network.

The convergence layer is further required to recognize the existence and timing of bandwidth grants, timeslots or channels associated with the transport of service and to map these services to the associated flows setup on the home network. Signalling protocol messages, derived from the external network, and associated with the setup of delivered services, can be used to derive the QoS parameters for the service on the home network. Flows set up on the home network, by the convergence layer, will be set up using QoS and timing information derived directly from the external network. More specifically, the convergence layer will direct the bandwidth manager in the G.9954 stack, to allocate TXOPs at a time within the synchronous MAC cycle that is closely synchronized with the bandwidth grants on the external network. This is used to control service latency and jitter.

Once the master is synchronized with the external network, and timeslots (TXOPs) have been synchronized with the arrival of data from the external network, the synchronization of endpoint devices follows naturally from the synchronous protocol defined for G.9954. Endpoints can synchronize with the (synchronized) master clock reference through its distribution of periodic Timestamp Reports or from information contained in the periodic MAP message. Furthermore, endpoints naturally synchronize on the timing of allocated TXOPs described in the MAP. The Event Indication mechanism can be used to notify the convergence layer at the endpoint of expected or granted TXOP timing information associated with a service. For flows that have their Timeslot

Event Indication flags enabled, the G.9954 MAC will notify (using interrupt or similar mechanisms) the upper convergence layer of the planned arrival of timeslot (TXOP) grants or service data. This indication can be used to drive a clock at the endpoint and/or to drive the data sampling rate at the endpoint.

It is still possible to synchronize to an external network without synchronizing clock references or sampling clocks. If clocks in the master and external network are not synchronized, a service may experience a MAXIMUM TRANSMISSION DELAY that is a function of the length of the MAC cycle accounting for worst-case acquisition period and network access latency. Furthermore, a lack of synchronization of the data (sample) arrival time and the allocated TXOP on the home network may result in the familiar saw-tooth latency/jitter behaviour as illustrated in Figure IV.2.

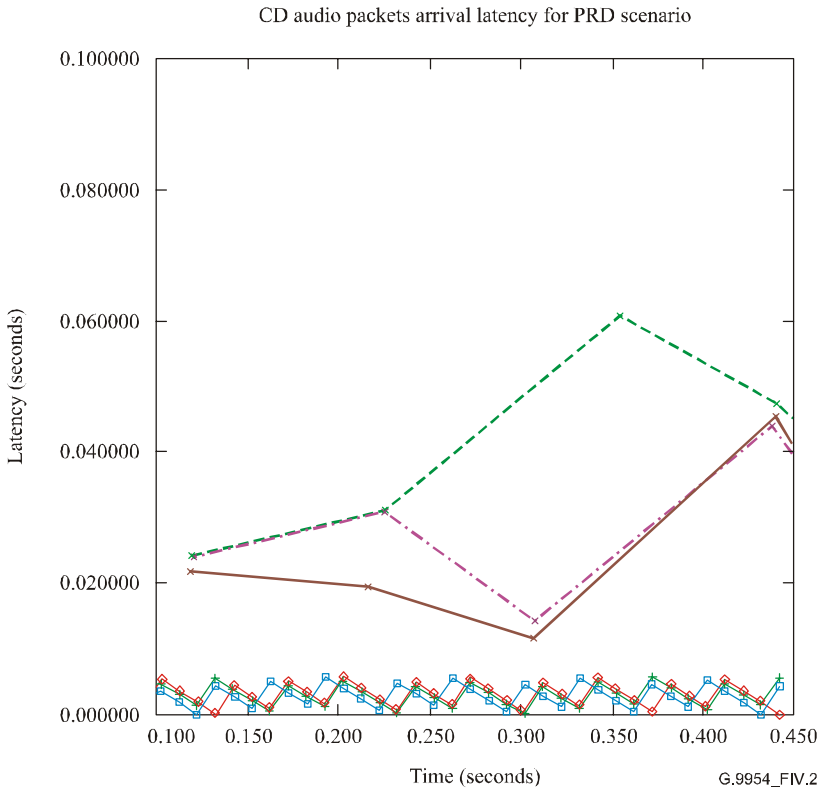


**Figure IV.2/G.9954 – Sawtooth latency/Jitter behaviour**

### IV.3 Summary of synchronization mechanisms

Table IV.I summarizes the set of synchronization mechanisms supported by the proposed G.9954 protocol.

**Table IV.1/G.9954 – Synchronization mechanisms summary**

| Synchronization mechanism | Purpose |
|---|---|
| Synchronous Protocol | Supports synchronization with other synchronous protocols. Endpoint synchronization with master |
| Clock Synchronization | Synchronize clocks to a common time reference. Synchronize sampling rates |
| MAC Cycle Indication | Synchronize external networks or protocols with MAC cycle |
| Timeslot Event Indications | Synchronize to planned TXOP timing using information in MAP |
| Timestamping Stream Data | Timestamp data using network clock reference |
| Timeslot Allocation Control | Synchronize the allocation of timeslots on the home network with timeslot grants in an external network |
| Protocol Convergence Layer | Supports protocol-specific handling of synchronization methods from external networks |

# Appendix V

# Support for Variable Bit Rate (VBR) flows

Variable Bit Rate (VBR) flows can be handled using the following different bandwidth allocation strategies:

- Per-Cycle Bandwidth Request;
- UGS + Shared Transmission Opportunity;
- UGS + Explicit Bandwidth Requests;
- UGS + Spare Bandwidth.

### V.1 Per-Cycle bandwidth request

This method requires that an explicit RTS request be issued each cycle. The amount of bandwidth requested each cycle is variable in accordance with the VBR behaviour of the service flow.

The bandwidth allocation method, although simple, may require some tight real-time control in order to ensure an endpoint node does violate of traffic rate characteristics and that QoS constraints can be met.

### V.2 UGS + shared transmission opportunity

The following method is suitable when there are several VBR flows active at one time. It is most appropriate when the source of all the VBR flows is the same station – i.e., there is no contention between VBR service flows, although it can also be used when the VBR flows originate from different stations.

This method assumes that a group of VBR flows will share the same transmission opportunity. The TXOP is allocated as for UGS service types (i.e., no explicit RTS is required); however, the amount

of bandwidth allocated is calculated to be the cumulative average bit rates of all the flows sharing the same TXOP.

The method relies on the variable nature of VBR flows. It assumes that VBR flows will NOT all peak at the same time, but rather the bandwidth demands of all VBR flows approximately equals their cumulative average.

This method is illustrated in Figure V.1.

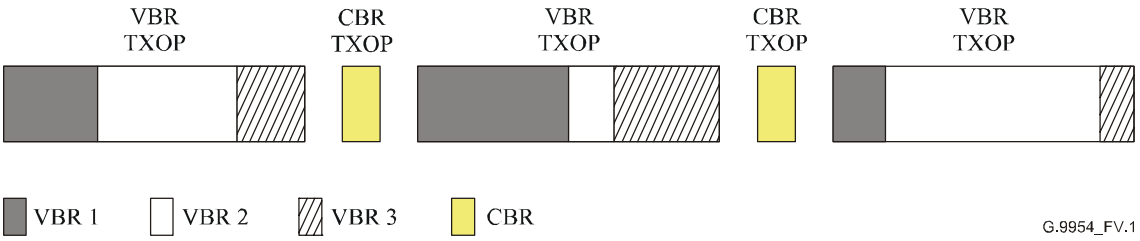


**Figure V.1/G.9954 – Variable Bit Rate (VBR) bandwidth allocation**

## V.3 UGS + explicit bandwidth requests

The following method (illustrated in Figure V.2) is a combination of the UGS and explicit bandwidth request methods. A VBR flow is treated like a CBR flow that may occasionally require some extra bandwidth to handle the variableness of the traffic. The basic data rate requested for the VBR flow is based on the flow's average bit-rate constraint.

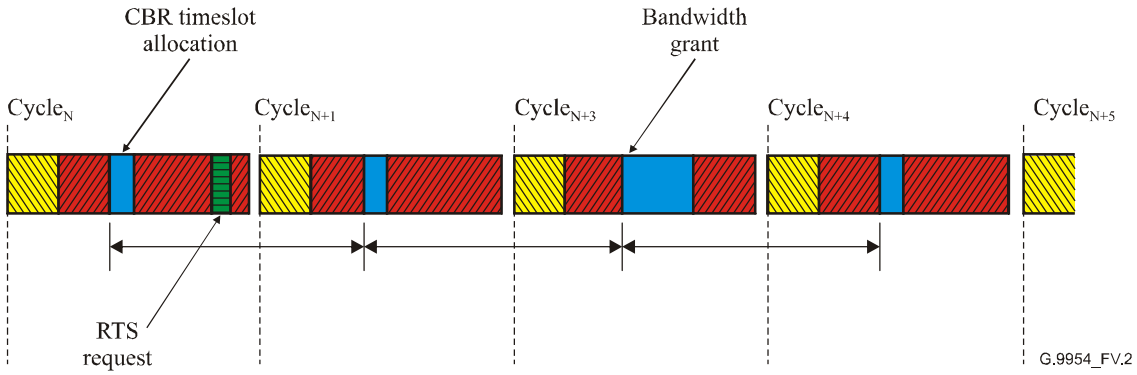


**Figure V.2/G.9954 – VBR using CBR + explicit bandwidth requests**

The allocation of fixed size TXOPs for a VBR flow effectively shapes the flow traffic into a Constant Bit Rate (CBR) form. If the flow has sufficient buffers associated with it to handle the burstiness of the traffic, this should be sufficient to handle the VBR nature of the flow without explicit bandwidth requests. However, if sufficient buffer space is NOT available, an endpoint node can explicitly request extra bandwidth in order to temporarily relieve the traffic backlog.

## V.4 UGS + spare bandwidth

Yet another method for handling VBR services involves using spare (unallocated) bandwidth to handle traffic bursts that exceed the traffic rate defined by the CBR TXOPs allocated for the flow. This is illustrated in Figure V.3.

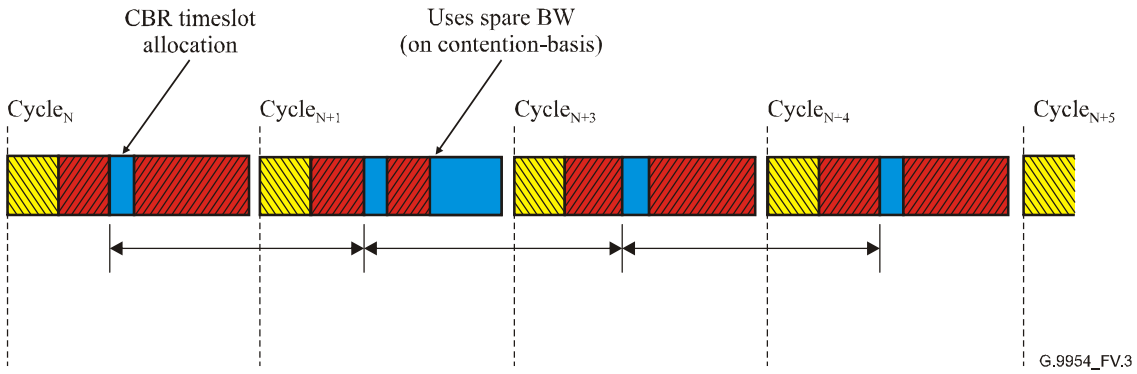


**Figure V.3/G.9954 – VBR using CBR + spare bandwidth**

Extra bandwidth may also be allocated to the TXOPs of a VBR service such that there is sufficient extra media time for the transmission of at least a whole (extra) packet. Using this method, the master scheduler should allocate a little more than the Average Bit Rate requirements, relying on the extra bandwidth to be used occasionally to empty traffic queues.

# Appendix VI

# Quality of Service (QoS) parameters

This Recommendation supports all services described in Table VI.1. In addition, this Recommendation should simultaneously support all services in Table VI.2.

**Table VI.1/G.9954 – Standard services QoS requirements[1]**

| Service | Relative priority | MAC payload rate (per stream) | Payload definition | Min. simultaneous streams | Max. bit error rate | Max. latency | Max. jitter |
|---|---|---|---|---|---|---|---|
| **Voice services** | | | | | | | |
| High quality narrowband voice telephony | High | 32-64 kbit/s | Voice payload[a)] | 8[b)] | 1e-6 | 5 ms nominal; 10 ms max | ±5 ms |
| Lower-quality narrowband voice telephony | Low to medium | 6-16 kbit/s | voice payload | 8 | 1e-6 | 10 ms nominal; 30 ms max | ±10 ms |
| Time-critical packet service (e.g., video conferencing) | High | 4-13 kbit/s for voice, 0.032-1.5 Mbit/s for audio/video | Voice payload for voice, MPEG-TS[c)] payload for audio/video | 4 (2 conversations 2 streams per conversation) | 1e-8 | 5 ms nominal; 10 ms max for full duplex services | ±5 ms |

1 Source: CableLabs "Home Networking Requirements for Cable-Based Services," Vendor Release 1.0 dated June 9, 2000. Copyright Cable Television Laboratories, Inc. 2001. All rights Reserved. Reprinted with permission (except as noted).

## Table VI.1/G.9954 – Standard services QoS requirements[1]

| Service | Relative priority | MAC payload rate (per stream) | Payload definition | Min. simultaneous streams | Max. bit error rate | Max. latency | Max. jitter |
|---|---|---|---|---|---|---|---|
| **High-speed data services** | | | | | | | |
| Best effort service | Low | Up to maximum physical layer rate | Data packet[d)] | N/A | 1e-6 | 500 ms | N/A |
| QoS (SLA[e)]) service | Medium to high | 10 Mbit/s | Data packet | 2 | 1e-8 | 10 ms nominal; 30 ms max | ±10 ms |
| **IP media streaming** | | | | | | | |
| Standard audio | Low to medium | 96-256 kbit/s | MPEG-TS | 3 | 1e-6 | 200 ms | ±20 ms |
| CD-quality audio | Medium | 192-256 kbit/s (stereo) | MPEG-TS | 3 | 1e-8 | 100 ms | ±10 ms |
| Lower-quality streaming video | Medium to high | 64-500 kbit/s | MPEG-TS | 3 | 1e-6 | 100 ms | ±10 ms |
| Home theater audio[f)] | High | 6 Mbit/s | MPEG-TS | 1 | 1e-8 | 100 ms | ±10 ms |
| Higher-quality streaming video | High | 1.5-10 Mbit/s | MPEG-TS | 1 | 1e-8 | 50 ms | ±10 ms |
| Digital Video Disk[g)] | | 3.0-20 Mbit/s | MPEG-TS | 2 | 1e-8 | 100 ms | ±10 ms |
| **Broadcast quality video** | | | | | | | |
| SDTV | High | 3-7 Mbit/s | | 2 | 1e-8 | 90 ms nominal | Interpacket ±10 ms |
| HDTV | High | 19.68 Mbit/s | | 1 | 1e-8 | 90 ms nominal | Interpacket ±10 ms |

a) Voice payload: variable size depending on codec, considering the end-to-end latency budget. For example, G.711 μ-law Encoding specifies frames of 4 samples where each audio sample is encoded as an 8-bit value (i.e., 32-bit).

b) The protocol developed for this technology must be able to support a minimum of 4 concurrent off-hook devices. With network connect rates greater than or equal to equivalent 10Base-T, the protocol shall support 8 concurrent off-hook devices.

c) MPEG-TS: Audio/Video payload assumes an MPEG Transport Stream (TS) of size 188 bytes.

d) Data packet: Ethernet payload including TCP/IP headers but excluding the Ethernet header and Ethernet CRC.

e) SLA used in this context implies "Service Level Agreement" and refers to a minimum Quality of Service that the service is committed to receiving. In this context, the SLA is taken to mean the "Committed Information Rate".

f) Home Theater Audio encompasses 5.1 channels of simultaneous audio. Note this is not included in the CableLabs document. It is assumed that the AC-3 Dolby Digital format is multiplexed in an MPEG-2 TS.

g) Digital Video Disk encompasses 2 SDTV streams. Note this is not included in the CableLabs document.

**Table VI.2/G.9954 – Additional standard services QoS requirements**

| Service | Relative priority | MAC payload rate (per stream) | Min. simultaneous streams | Max. bit error rate | Max. latency | Max. jitter |
|---|---|---|---|---|---|---|
| **Voice services** | | | | | | |
| High-quality Narrowband Voice Telephony | High | 32-64 kbit/s | 6<br>(3 conversations; 2 streams per conversation) | 1e-6 | 5 ms nominal; 10 ms max. | ±5 ms |
| Time-critical packet service (e.g., video conferencing) | High | 4-13 kbit/s for voice; 0.032-1.5 Mbit/s for audio/video | 2<br>(1 conversation; 2 streams per conversation) | 1e-8 | 5 ms nominal; 10 ms max. for full-duplex services | ±5 ms |
| **High-speed data services** | | | | | | |
| Best effort service | Low | Up to maximum physical layer rate | N/A | 1e-6 | 500 ms | N/A |
| **IP media streaming** | | | | | | |
| CD-quality audio | Medium | 192-256 kbit/s (stereo) | 3 | 1e-8 | 100 ms | ±10 ms |
| *Any 2-stream combinations of the following:* | | | | | | |
| Higher-quality streaming video | High | 1.5-10 Mbit/s | 1 | 1e-8 | 50 ms | ±10 ms |
| Home theater audio | High | 6 Mbit/s | 1 | 1e-8 | 100 ms | ±10 ms |
| Digital Video Disk | | 3.0-20 Mbit/s | 1 | 1e-8 | 100 ms | ±10 ms |
| **Broadcast-quality video** | | | | | | |
| SDTV | High | 3-7 Mbit/s | 2 | 1e-8 | 90 ms nominal | Interpacket ±10 ms |
| HDTV | High | 19.68 Mbit/s | 1 | 1e-8 | 90 ms nominal | Interpacket ±10 ms |

# Appendix VII

# Simultaneous applications test profiles

Test profile 1 describes a home network consisting of a Residential Gateway (RG) providing access to telephony and Internet services and a second Gateway or Server source providing access to video-related services. The RG and possibly the Video/TV Server are connected to broadband pipes. Furthermore, the home network profile consists of clients that consume broadband services as well as those that interact directly with peers on the same home network.

The network throughput requirements for test profile 1 are described in Table VII.1. It is assumed the network configuration is a star network, with 6-foot wire attached to each.

**Table VII.1/G.9954 – Network throughput requirements**

| Service | Quantity | Rate [Mbit/s] | Throughput requirement |
|---|---|---|---|
| HQ Voice | 6 | 0.064 | 0.384 |
| Video conference | 2 | 1.5 | 3 |
| Best effort | 1 | Up to physical limit | Up to physical limit |
| CD | 3 | 0.256 | 0.768 |
| SDTV | 2 | 3 | 6 |
| HDTV | 1 | 19.68 | 19.68 |
| Home theater | 2 | 5.76 | 11.52 |
| TOTAL | | | 41.352 |

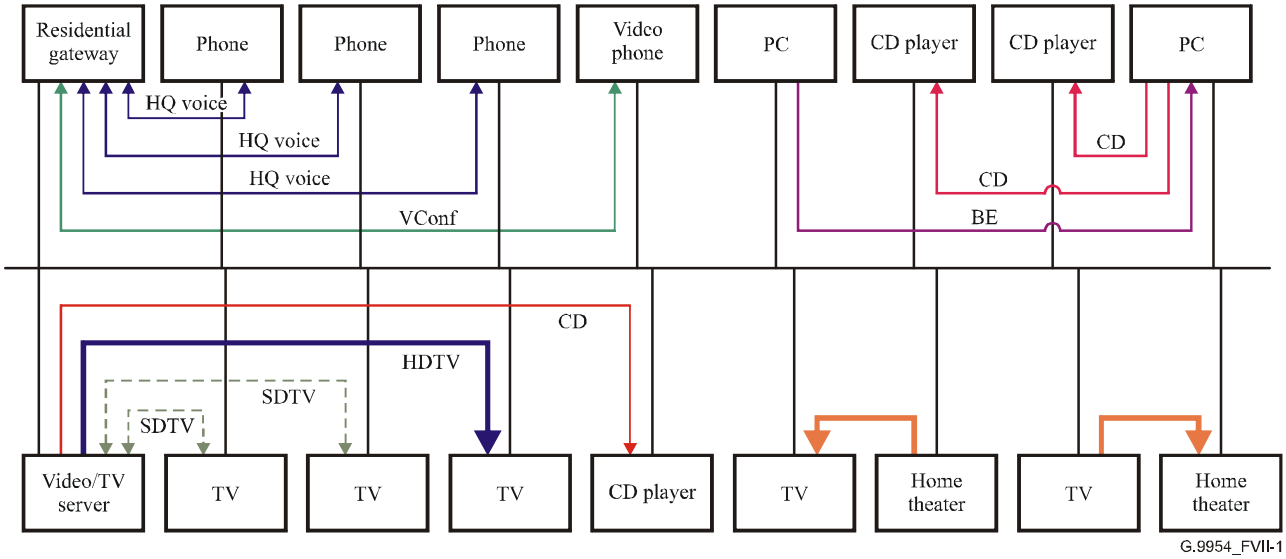


**Figure VII.1/G.9954 – Test profile 1**

# Appendix VIII

## Media access planning guidelines

Media access planning is a scheduling activity whose goal is to produce a Media Access Plan (MAP) that satisfies the QoS constraints of all contending flows in the network. The scheduling algorithm executes entirely in the master node and takes into consideration the available media bandwidth and the QoS constraints of the entire network.

Although the specification of scheduling algorithms employed by a G.9954 master is beyond the scope of this Recommendation it is expected that a G.9954 master scheduler support the following set of basic functional capabilities:

- Resource management;
- Media resource allocation and assignment;
- Burst size management;
- MAC cycle length management;

- Traffic policing and shaping;
- Latency and jitter control;
- Collision management strategy assignment;
- Bandwidth request management;
- MAP generation.

### VIII.1 Resource management

The master should manage state information about the allocation of media resources in the home network and maintain an allocation map that describes allocated and free media resources and their sizes. The allocation map is used by the Bandwidth Allocation function when performing admission control for service requests.

### VIII.2 Media resource allocation and assignment

Given the availability of sufficient media resources to service a bandwidth request, the master should allocate TXOPs to the specific flow. The allocated TXOP is subsequently described in the MAP.

### VIII.3 Burst size management

In order to use the media more efficiently and to reduce protocol overheads, it is desirable to aggregate upper-level packets originating from a single source or flow into single PHY layer bursts (frames). The length of the burst depends on a number of factors including the length of the TXOP, flow latency requirements, BER characteristics, etc.

The master scheduler should try to concentrate TXOPs assigned to the same source such that an endpoint can maximize the length of the bursts while still meeting flow QoS latency and jitter constraints.

### VIII.4 MAC cycle length management

Each MAP frame implicitly defines the extent (in time) of the Media Access Plan. This provides the infrastructure to support MAC cycles that are variable in length and that may even change dynamically from cycle to cycle.

The master scheduler is responsible for selecting the appropriate size of the MAC cycle. The guidelines used in the selection process require the scheduler to select a cycle length that balances the periodicity requirements of the active flows with protocol overhead considerations introduced by the transmission of the MAP frame.

### VIII.5 Traffic policing and shaping

To ensure the conformance of a flow to its negotiated traffic parameters, the master scheduler should police and shape traffic such that the network will not suffer in case a traffic source starts to generate traffic in a non-conformant manner. Traffic policing and shaping is done by allocating TXOPs in a manner that meets traffic specifications.

For a G.9954 endpoint node that assigns packets to TXOPs in accordance with the description of the MAP, this will inherently shape the endpoints traffic into the form intended by the master. This has the effect of reducing the potential complexity of endpoint nodes by centralizing the traffic policing and shaping algorithms in the master while also ensuring that endpoint nodes do not generate traffic in a manner that violates their negotiated agreement.

## VIII.6 Latency and jitter control

The master scheduler is responsible for performing latency and jitter control by guaranteeing that TXOPs are allocated to flows at the required frequency, size and interval that allows them to meet flow latency and jitter requirements.

Consider two examples allocations of TXOPs over time (in Figure VIII.1), relative to the arrival time of the packets from the input source. In example 1, TXOPs are allocated such that they provide zero jitter. In example 2, the latency variance causes jitter as seen in Figure VIII.2.

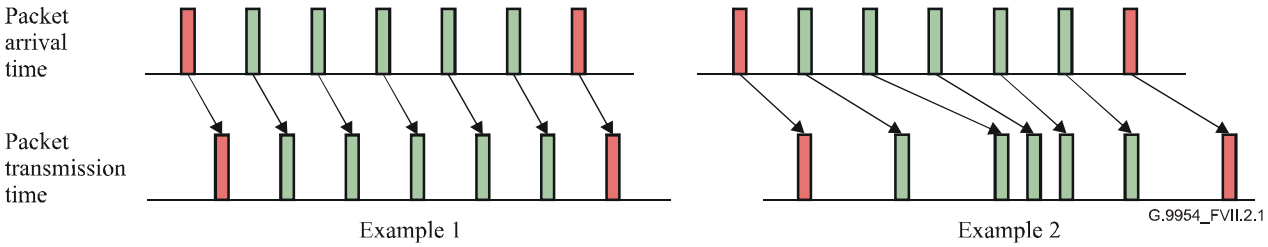


**Figure VIII.1/G.9954 – Latency/jitter examples**

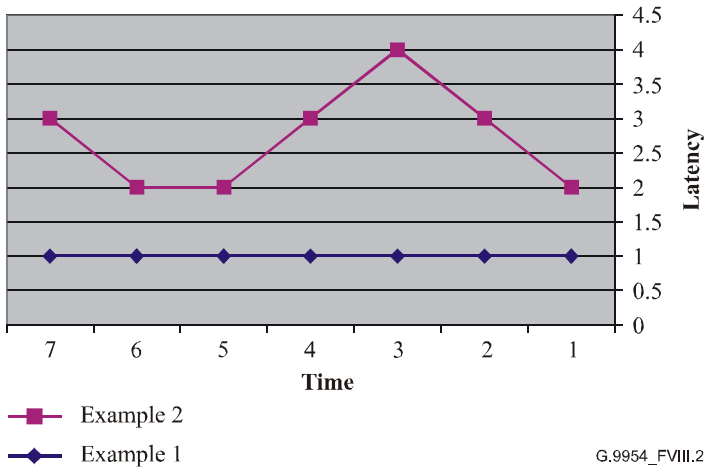


**Figure VIII.2/G.9954 – Latency/jitter graph**

## VIII.7 MAP generation

The output of the master's media access planning is the MAP frame. The master is responsible for generating the periodic MAP control frame that contains the results of the processes and decisions described above.

# BIBLIOGRAPHY


[4] Data-Over-Cable Service Interface Specifications – *Cable Modem to Customer Premise Equipment Interface Specification SP-CMCI-I05-001215, July 14, 2000*.

[5] Data-Over-Cable Service Interface Specifications – *Radio Frequency Interface Specification, SP-RFIv1.1-I06-001215, December 15, 2000*.

[6] P1394.1 *Draft Standard for High Performance Serial Bus Bridges, 0.16, March 29, 2001*.

## SERIES OF ITU-T RECOMMENDATIONS

| | |
|---|---|
| Series A | Organization of the work of ITU-T |
| Series D | General tariff principles |
| Series E | Overall network operation, telephone service, service operation and human factors |
| Series F | Non-telephone telecommunication services |
| **Series G** | **Transmission systems and media, digital systems and networks** |
| Series H | Audiovisual and multimedia systems |
| Series I | Integrated services digital network |
| Series J | Cable networks and transmission of television, sound programme and other multimedia signals |
| Series K | Protection against interference |
| Series L | Construction, installation and protection of cables and other elements of outside plant |
| Series M | Telecommunication management, including TMN and network maintenance |
| Series N | Maintenance: international sound programme and television transmission circuits |
| Series O | Specifications of measuring equipment |
| Series P | Telephone transmission quality, telephone installations, local line networks |
| Series Q | Switching and signalling |
| Series R | Telegraph transmission |
| Series S | Telegraph services terminal equipment |
| Series T | Terminals for telematic services |
| Series U | Telegraph switching |
| Series V | Data communication over the telephone network |
| Series X | Data networks, open system communications and security |
| Series Y | Global information infrastructure, Internet protocol aspects and next-generation networks |
| Series Z | Languages and general software aspects for telecommunication systems |

Printed in Switzerland
Geneva, 2006