Working Draft

T11/Project 1237-D/Rev 0.8

February 10, 1998

Information Technology - Fibre Channel - Audio Video (FC-AV)

Draft proposed American National Standard (dpANS)
Secretariat: National Committee for Information Technology Standardization (NCITS)

This is a draft proposed American National Standard of Accredited Standards Committee NCITS. As such, this is not a completed standard. The T11 Technical Committee may modify this document as a result of comments received during public review and its approval as a standard.

ABSTRACT:

This document describes the mapping to Fibre Channel of certain audio and video digital formats such as those used in digital television and those described by MPEG and related compression standards. It also describes a recommended practice for synchronizing applications of Fibre Channel within the broadcast environment.

Facilitator:
Bob Edge
Tektronix, Inc.
14180 SW Karl Braun Drive
P.O. Box 500, M/S 58-850
Beaverton, OR 97077

Voice: (503) 627-6210
FAX: (503) 627-1137
E-Mail: bob.c.edge@tek.com

Technical Editor:
Robert S. Cornelius
Ancor Communications, Inc.
6130 Blue Circle Drive
Minnetonka, MN 55343

Voice: (612) 932-4022
FAX: (612) 932-4037
E-Mail: bobc@ancor.com

Reference number
ISO/IEC ***** : 199x
NCITS, *** - 199x
Printed 2/10/1998

Other Points of Contact:

T11 Chairman:

Roger Cummings
Distributed Processing Technology
140 Candace Drive
Maitland, FL 32751

Voice: (407) 830-5522 x348
FAX: (407) 260-5366
E-Mail: cummings_roger@dpt.com

T11 Vice-Chairman:

Edward L. Grivna
Cypress Semiconductor
2401 East 86th Street
Bloomington, MN 55425

Voice: (612) 851-5046
FAX: (612) 851-5087
E-Mail: elg@cypress.com

NCITS Secretariat:

NCITS Secretariat, ITI
1250 Eye Street, NW Suite 200
Washington, DC 20005

Voice: (202) 737-8888
FAX: (202) 638-4922
E-Mail: ncitssec@itic.nw.dc.us

Document Distribution:

Global Engineering Documents
15 Inverness Way East
Englewood, CO 80112-5776

Voice: (303) 397-7956
800: (800) 854-7179
FAX: (303) 397-2740

T11 Reflector (for minutes, agendas, etc.)

Internet address for subscription to the T11 reflector:	Majordomo@nsco.network.com
Note should contain a line stating...	subscribe T11 *your email address*
Internet address for distribution via T11 reflector:	t11@nsco.network.com

Technical Reflector (for technical discussions)

Internet address for subscription to the FC reflector:	Majordomo@nsco.network.com
Note should contain a line stating...	subscribe fc *your email address*
Internet address for distribution via FC reflector:	fc@nsco.network.com

Web sites:	http://www.ncits.org/t11 http://www.dpt.com/t11

PATENT STATEMENT

CAUTION: The developers of this standard have requested that holder's of patents that may be required for the implementation of the standard, disclose such patents to the publisher. However, neither the developers nor the publisher have undertaken a patent search in order to identify which, if any, patents may apply to this standard.

As of the date of publication of this standard and following calls for the identification of patents that may be required for the implementation of the standard, no such claims have been made. No further patent search is conducted by the developer or the publisher in respect to any standard it processes. No representation is made or implied that licenses are not required to avoid infringement in the use of this standard.

Comments on Rev 0.8

This is a preliminary document that may change substantially prior to completion. It contains a collection of submissions that have not been reviewed by the full FC-AV working group.

Changes are marked with change bars in the margins.

The *FC-AV Transport Container/Header January 1998* document prepared by Michael W. Pugh has been inserted as clause 10.

Contents

Page

Page

Page

Page

draft proposed American National Standard
for Information Technology—

Fibre Channel — Audio-Video (FC-AV)

1 Scope

This draft American National Standard specifies the transport of existing representations of audio and video information over Fibre Channel and the interoperation of digital segments based on Fibre Channel with other analog and digital equipment.

Specifications are included for:

- the mappings to Fibre Channel of the formats defined by the ITU-R BT-601 digital television standard and its derivatives;
- the mappings to Fibre Channel of the formats defined by the ISO/IEC 13818 family of standards which include MPEG and related compression systems;
- recommended practice for the synchronization of a Fibre Channel segment with existing analog and digital segments;
- a coherent framework for the mappings to Fibre Channel of other and future audio and video representations;
- the mappings to Fibre Channel of existing studio machine control methodologies.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standards listed below.

Copies of the following documents can be obtained from ANSI: Approved ANSI standards, approved and draft international and regional standards (ISO, IEC, CEN/CENELEC, ITUT), and approved and draft foreign standards (including BSI, JIS, and DIN). For further information, contact ANSI Customer Service Department at 212-642-4900 (phone), 212-302-1286 (fax) or via the World Wide Web at http://www.ansi.org.

2.1 Approved references

ANSI X3.230-1994, *Fibre Channel–Physical and Signaling Interface (FC-PH)*

.ANSI X3.297-1997, *Fibre Channel–Physical and Signaling Interface-2 (FC-PH-2).*

ANSI X3.272-1996, *Fibre Channel--Arbitrated Loop (FC-AL).*

ANSI X3.269-1996, *SCSI-3 Fibre Channel Protocol (FCP).*

ANSI/SMPTE 125M-1992, *Television - Component Video Signal 4:2:2 Bit-Parallel Interface.*

ANSI/SMPTE 259M-1993, *Television - 10-Bit 4:2:2 Component and 4fsc NTSC Composite Digital Signals - Serial Digital Interface.*

International Telecommunication Union Recommendation ITU-R BT.601-4, *Encoding Parameters of Digital Television for Studios.*

2.2 References under development

At the time of publication, the following referenced standards were still under development. For information on the current status of the documents, or regarding availability, contact the relevant standards body or other organization as indicated.

ANSI X3.303-199x, *Fibre Channel–Physical and Signaling Interface-3 (FC-PH-3)*[1].

3 Definitions and conventions

For the purpose of dpANS X3.xxx, FC-AV, the following definitions, conventions, abbreviations, and acronyms apply.

3.1 Definitions

3.1.1 active video: That portion of a complete video scan that is visible on the display screen.

3.1.2 Begin Stream: An Extended Link Service command which enables a Class 4 Stream Definer to activate a Class 4 Stream (see 6.2.1).

3.1.3 chrominance: The color portion of a video signal. Chrominance includes hue and saturation information but not brightness.

3.1.4 Class 4 Asynchronous Stream: A Class 4 Stream, consisting of a single Component Stream, which honor only its Maximum Class 4 Stream Bandwidth limit. However, a Minimum Class 4 Stream Bandwidth limit, sufficient to prevent timeout, due to the E_D_TOV limit, is maintained.

3.1.5 Class 4 Available Bit Rate Stream: A Class 4 Stream, containing one or more Component Streams, each which honor only its Maximum Class 4 Stream Bandwidth limit. However, a Minimum Class 4 Stream Bandwidth limit, sufficient to prevent timeout, due to the E_D_TOV limit, is maintained.

3.1.6 Class 4 Get Tag: An Extended Link Service command which enables a Class 4 Stream Definer to obtain a Stream Tag for a new Class 4 Stream, and the System Resource Manager to obtain a Producer Tag for a new Class 4 Stream (see 6.2.2).

3.1.7 Class 4 Real–Time Stream: A Class 4 Stream which honor both the Maximum Class 4 Stream Bandwidth limit and the Minimum Class 4 Stream Bandwidth limit.

3.1.8 Class 4 Stream: 1. A Class 4 process whereby Class 4 circuits are established and removed, at the request of a Class 4 Stream Definer, through an intermediary, the System Resource Manager, which allocates Class 4 Stream Resources, not only for the Fabric but also for both the source of the Class 4 Stream, the Class 4 Stream Producer, and the destination of the Class 4 Stream, the Class 4 Stream Consumer. 2. A generic reference to any of the following; Class 4 Asynchronous Stream, Class 4 Available Bit Rate Stream or Class 4 Real–Time Stream.

3.1.9 Class 4 Stream circuit: A Class 4 circuit, as defined by FC–PH–2, but managed by the System Resource Manager, rather than the Quality of Service Facilitator.

3.1.10 Class 4 Stream circuit Control: An Extended Link Service command which enables the System Resource Manager to establish or modify a Class 4 Stream circuit (see 6.2.3).

3.1.11 Class 4 Stream circuit End: An Extended Link Service command which enables the Quality of Service Facilitator to inform the System Resource Manager of the removal, release, of a Class 4 Stream circuit (see 6.2.4).

3.1.12 Class 4 Stream Consumer: The Fibre Channel Port terminating a Class 4 Stream (sink).

3.1.13 Class 4 Stream Definer: An entity ini-

---

[1] For information about obtaining copies of this document or for more information on the current status of the document, contact the X3 Secretariat at http://www.x3.org or 202-626-5783.

tiating the establishment, suspension, resumption and termination of Class 4 Streams.

3.1.14 Class 4 Stream Notify: An Extended Link Service command which the System Resource Manager uses to inform the Class 4 Stream Consumer of a new Class 4 Stream (see 6.2.5).

3.1.15 Class 4 Stream Pacing: 1. A method whereby Class 4 frame flow on a Class 4 Stream is bandwidth limited. 2. A generic reference to any of the following; Consumer Pacing, Fabric Pacing or Producer Pacing.

3.1.16 Class 4 Stream Port: A Fibre Channel Ports supporting Class 4 Streams, able to function as one or more of the following; Class 4 Stream Consumer, Class 4 Stream Definer, Class 4 Stream Producer or System Resource Manager.

3.1.17 Class 4 Stream Producer: The Fibre Channel Port originating a Class 4 Stream (source).

3.1.18 Class 4 Stream Resource: A generic reference to any of the following; Maximum Class 4 Stream Bandwidth limit, Minimum Class 4 Stream Bandwidth limit, Maximum Class 4 Stream Latency, Stream Concurrent Sequences, Stream End–to–End Credit or Stream Data Field Size.

3.1.19 Class 4 Stream virtual circuit: An unidirectional Class 4 virtual circuit, as defined by FC–PH–2, i.e. one half of a Class 4 Stream circuit.

3.1.20 Composite Class 4 Stream circuit: A Class 4 Stream circuit able of supporting multiple Class 4 Streams concurrently.

3.1.21 Component Stream: Part of a Class 4 Stream, carrying a specific portion, component, of the information flowing from the Class 4 Stream Producer to the Class 4 Stream Consumer. A Component Stream generally flow on a single unique Exchange, supporting zero, one or more concurrent Sequences.
Component Streams uses the following Class 4 Stream Resources; Maximum Class 4 Stream Bandwidth limit, Minimum Class 4 Stream Bandwidth limit, Stream End–to–End Credit and Stream Concurrent Sequences.

3.1.22 component video: The unencoded output of a camera or other device consisting of three primary color signals: red, green, and blue (RGB) that together convey all necessary picture information.

3.1.23 composite video: An encoded video signal, such as NTSC or PAL video, that includes horizontal and vertical synchronizing information.

3.1.24 Consumer Pacing: A method of flow control for a Class 4 Stream, which rely on the Class 4 Stream Consumer to control the ACK frame flow in order to honor either or both the Maximum Class 4 Stream Bandwidth limit and the Minimum Class 4 Stream Bandwidth limit.

3.1.25 Consumer VC_ID: A Class 4 Stream Resource, assigned by the System Resource Manager, to the Class 4 Stream virtual circuit, in the direction from the Class 4 Stream Consumer to the Class 4 Stream Producer.

| **3.1.26 Container:**

3.1.27 Conventional Class 4 circuit: A Class 4 circuit, as defined by FC–PH–2, which is not managed by the System Resource Manager.

3.1.28 Fabric Pacing: A method of flow control for a Class 4 Stream, which rely on the Fabric to control the VC_RDY flow in order to honor either or both the Maximum Class 4 Stream Bandwidth limit and the Minimum Class 4 Stream Bandwidth limit.

3.1.29 field: Half of the interlaced horizontal lines needed to create a complete picture. Two interlaced fields create a complete frame or picture.

3.1.30 frame: A complete video picture composed of two complete interlaced scans of the monitor screen.

3.1.31 Founding Stream Consumer: The Class 4 Stream Port which was the Class 4 Stream Consumer when the Class 4 Stream circuit initially was established.

3.1.32 Founding Stream Producer: The Class 4 Stream Port which was the Class 4 Stream Producer when the Class 4 Stream circuit initially was established.

3.1.33 Get Class 4 Stream Ability: An Extended Link Service command, which lets a Class 4 Stream Port obtain the Class 4 Stream characteristics of another Class 4 Stream Port, and lets the System Resource Manager obtain the Class 4 Stream characteristics of both another Class 4 Stream Port and for a Switch Port (E_Port or F_Port). (See 6.2.6)

3.1.34 interlaced scanning: A system of video scanning whereby the odd and even numbered lines of a picture are transmitted consecutively as two separate interleaved fields.

3.1.35 luminance: That part of a video signal which carries the brightness and contrast information.

3.1.36 Maximum Class 4 Stream Bandwidth limit: The largest amount of bandwidth, in units of Bytes per second, available to either a Class 4 Stream[2] or a Component Stream.

3.1.37 Maximum Class 4 Stream Latency: The longest time, in units of 1/65 536 second, that a Class 4 frame shall spend in the Fabric.

3.1.38 Minimum Class 4 Stream Bandwidth limit: The smallest amount of bandwidth, in units of Bytes per second, available to either a Class 4 Stream2) or a Component Stream.

| **3.1.39 Object:**

3.1.40 Permanent Stream: A fixed Class 4 Stream circuit carrying a Class 4 Asynchronous Stream in both directions, between Class 4 Stream Ports and either the Quality of Service Facilitator (Permanent Stream #1) or the System Resource Manager (Permanent Stream #2).

3.1.41 Producer Tag: An unique Class 4 Stream Producer assigned identifier, which in conjunction with the Class 4 Stream Producer Identifier, distinguish a particular Class 4 Stream.

3.1.42 Producer VC_ID: A Class 4 Stream Resource, assigned by the System Resource Manager, to the Class 4 Stream virtual circuit, in the direction from the Class 4 Stream Producer to the Class 4 Stream Consumer.

3.1.43 Producer Pacing: A method of flow control for a Class 4 Stream, which rely on the Class 4 Stream Producer to control the Data Frame flow in order to honor either or both the Maximum Class 4 Stream Bandwidth limit and the Minimum Class 4 Stream Bandwidth limit.

3.1.44 Query Tag: An Extended Link Service command enabling a Class 4 Stream Definer and the System Resource Manager to obtain a list of Tags assigned to a given Class 4 Stream Producer (see 6.2.7).

3.1.45 Real Time:

3.1.46 Register System Resource Manager: An Extended Link Service command enabling a Class 4 Stream Port to become System Resource Manager (see 6.2.8).

3.1.47 Stream Concurrent Sequences: A Class 4 Stream Resource, specifying the largest number of Sequence that may be Open for a Component Stream.

3.1.48 Stream Consumer: That logical entity that receives a data Stream.

3.1.49 Stream Data Field Size: A Class 4 Stream Resource, specifying the largest Data Field Size (see FC–PH clause 17) for any Class 4 Data Frame sent on a given Component Stream.

3.1.50 Stream Definer: That logical entity that initiates the establishment, suspension, resumption, and termination of data Streams.

3.1.51 Stream End–to–End Credit: A Class 4 Stream Resource, specifying the largest number of unacknowledged Data Frames a Class 4 Stream Producer may have outstanding for a Component Stream.

3.1.52 Stream Producer: That logical entity that sources a data Stream.

3.1.53 Stream Tag: An unique System Resource Manager assigned identifier, which in conjunction with the Class 4 Stream Producer Identifier, distinguish a particular Class 4 Stream.

3.1.54 Stream Type: A generic reference to any of the following; Class 4 Asynchronous Stream, Class 4 Available Bit Rate Stream or Class 4 Real–Time Stream.

3.1.55 Suspend Stream: An Extended Link Service command which enables a Class 4 Stream Definer to deactivate a Class 4 Stream (see 6.2.9).

---

2) The Bandwidth applies to transfers for which an average Frame Size is equal to the Stream Data Field Size, averaged over 1 ms.

3.1.56 System Resource Manager: An entity, with a Class 4 Stream Port, responsible for the allocation and release of all of the Class 4 Stream Consumers, Class 4 Stream Producers and the Fabric Class 4 Stream Resources.

3.1.57 Tag Query: An Extended Link Service command which enables a Class 4 Stream Definer and the System Resource Manager to discover the characteristics of a specific Class 4 Stream (see 6.2.10).

3.1.58 Terminate Stream: An Extended Link Service command which enables a Class 4 Stream Definer to terminate a Class 4 Stream (see 6.2.11).

3.1.59 YUV: A color encoding scheme in which luminance and chrominance are separate.

3.2 Editorial conventions

In dpANS X3.xxx, FC-AV, a number of conditions, mechanisms, sequences, parameters, events, states, or similar terms are printed with the first letter of each word in uppercase and the rest lowercase (e.g., Fabric, Class). Any lowercase uses of these words have the normal technical English meanings.

Numbered items in dpANS X3.xxx, FC-AV do not represent any priority. Priority is explicitly indicated.

In case of any conflict between figure, table, and text, the text takes precedence. Exceptions to this convention are indicated in the appropriate sections.

In all of the figures, tables, and text of this document, the most significant bit of a binary quantity is shown on the left side. Exceptions to this convention are indicated in the appropriate sections.

The term “shall” is used to indicate a mandatory rule. If such a rule is not followed, the results are unpredictable unless indicated otherwise.

If a field or a control bit in a frame is specified as not meaningful, the entity which receives the frame shall not check that field or control bit.

3.3 Abbreviations and acronyms

Abbreviations and acronyms applicable to this standard are listed. Definitions of several of these items are included in 3.1”Definitions”.

ABR	Available bit rate
FC-AL	See 2.1
FCP	See 2.1
FC-PH	See 2.1
FC-PH-2	See 2.1
FC-PH-3	See 2.2
FFT	Fast File Transfer Protocol (see clause 9)
NTSC	National Television Systems Committee
PAL	Phase Alternate Line
RT	Real Time
SCSI	Small Computer System Interface

4 Structure and concepts

This clause provides an overview of the structure, concepts and mechanisms used in FC-AV and is intended for informational purposes only.

4.1 Relationship with FC-PH

ANSI X3.230-1994, the Fibre Channel - Physical and Signalling Interface standard (FC-PH), together with its extensions FC-PH-2 and FC-PH-3, describe a high-performance, transport vehicle for Upper Level Protocols (ULPs). This document, FC-AV, describes standard methods, applications, and services that enable audio, video, ancillary data and control streams to use FC-PH and its extensions as a transport service.

The Fibre Channel standards are organized in a layered manner illustrated in figure 1. The highlighted portion of figure 1 is the subject of this document.

The four highlighted vertical columns describe two methods for file transfer:

- FTP/TCP/IP
- Fast file transfer (FFT)

and two methods for real-time stream transfer:

- Sender push model
- Receiver pull model.

4.1.1 File transfers

In the context of this document, files are distinct from streams in that the material being transferred, as a file, is both fixed and limited in length.

4.1.1.1 Legacy method file transfers

FC-AV recommends the FTP / TCP / IP protocol suite for file transfer as this method of file transfer is widely supported across multiple vendor platforms.

FTP, while highly interoperable, is not generally optimized for either high-bandwidth data transfers or for large file transfers.

4.1.1.2 High-performance file transfers

Within the framework of FC-AV, high performance means that material is able to be transferred faster than in real time, i.e. that the material is transferred faster than it would normally be viewed.

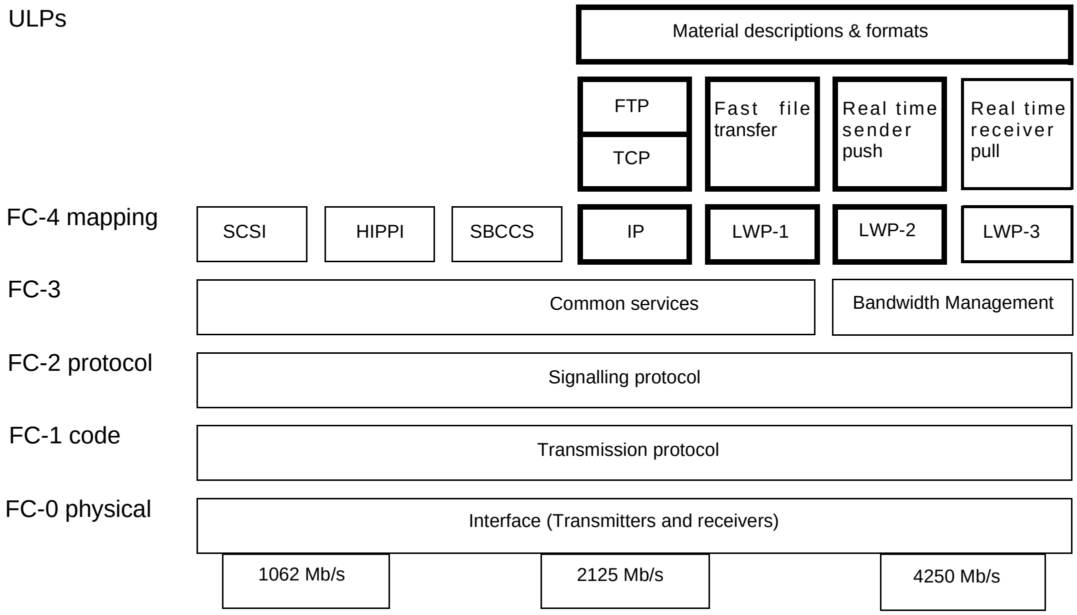

Figure 1 – FC-AV Model

Clause 9 describes the FC-AV Fast File Transfer Protocol (FFT). FFT is a hybrid protocol stack. It uses TCP/IP to set up and tear down the channel being used for data movement and it uses FCP, the SCSI transport protocol for Fibre Channel, to actually transport the data.

4.1.2 Stream transfers

Within the framework of FC-AV, streams transfer material of unknown length in apparent real time or precise multiples of real time.

The following three types of Streams are defined:

– *Asynchronous Stream:* Used to carry intermittent, sporadic information between a Stream Producer and a Stream Consumer;

– *Available Bit Rate (ABR) Stream:* Used to carry frequent or recurring information, at a rate governed by the System Resource Manager, the Stream Producer and, for Class 2, the Stream Consumer, on one or more Exchanges between a Stream Producer and a Stream Consumer;

– *Real–Time Stream:* Used to carry Real–Time information, with guaranteed Quality of Service parameters, on one or more Exchanges between a Stream Producer and a Stream Consumer.

4.1.2.1 Push model real-time streams

4.1.2.2 Pull model real-time streams

4.2 Bandwidth management

Methods are described for providing bandwidth allocation and management for data streams using various Classes of Fibre Channel services.

4.2.1 System Resource Manager

The System Resource Manager is the logical entity that reserves bandwidth for Stream Producers and Stream Consumers at the request of a Stream Definer.

Access to the System Resource Manager is obtained through a set of Extended Link Services.

4.2.2 Class 2 and Class 3 Streams

Class 2 and Class 3 Streams (see clause 5) is an enhancement to existing Fibre Channel Class 2 and Class 3 services (see FC-PH). It is based on dividing each second of data transfer at 1.0625 GBaud into 64 fixed time slots. Higher Fibre Channel speeds support a proportionally larger number of slots.

Stream Control Services (SCS) are provided by a System Resource Manager. A Stream Definer acquires one or more time slots from the System Resource Manager sufficient to support Real-Time or Available Bit Rate Streams. For example, 30 time slots are assigned to carry one 270 Mbit/sec Stream, and 60 time slots are assigned to carry one 360 Mbit/sec Stream.

4.2.3 Class 4 Streams

Class 4 Streams (see clause 6) are carried on Class 4 circuits (see FC-PH-2)., here known as Class 4 Stream circuits. Each Class 4 Stream circuit consists of counter-directional virtual circuits, each with an individually defined set of Quality of Service parameters.

In Class 4 operation, a Quality of Service Facilitator (QoSF) manages the guaranteed bandwidth service. In Class 4 Stream operation, a System Resource Manager works in conjunction with the Quality of Service Facilitator to provide Class 4 Stream services.

4.2.4 Class 6 Multicast Streams

Class 6 Multicast Streams (see clause 7) is an enhancement to existing Fibre Channel Class 6 Multicast services (see FC-PH-3).

The Class 6 Multicast streaming function requires an agreement among user nodes to abide by a set of streaming rules within a multicast group. The Fabric provides Class 6 services but does not provide enforcement of those rules required to maintain a streaming function. Other Classes of Service may coexist with Class 6 service on a single Fabric.

Stream Control Services (SCS) are provided by a System Resource Manager. A Stream Definer acquires one or more time slots from the System Resource Manager sufficient to support Real-Time or Available Bit Rate Streams.

4.2.5 Application Interface (API)

A common application interface (API) for bandwidth management is described in clause 8.

4.3 FC-AV container system

The FC-AV container system (see clause 10) is designed to optimize the transport of video data over Fibre Channel. An essential element of the optimization is the grouping of video, audio and ancillary data into relatively large data sets to be transported as a unit. Using 4:2:2 uncompressed video as an example, the data set is of the order of 1 megabyte of data per video frame. Fibre Channel is well suited to block transmission of such a large data set, and is much more efficient when data is grouped in large blocks rather than small packets.

4.4 Television video

A primer on television video is presented in Annex A. A summary of digital video formats for television is given in table 1.

Table 1 – Digital Video Formats for Television

Format	4:2:2 - 18MHz		4:2:2 - 13.5 MHz		PAL 625	NTSC 525
Bit Rate	360 Mb/s	360 Mb/s	270 Mb/s	270 Mb/s	177 Mb/s	143 Mb/s
Lines/Frame	625	525	625	525	625	525
Field Rate	50 /s	59.94 /s	50 /s	59.94 /s	50 /s	59.94 /s
Sample Rate	18 MHz	18 MHz	13.5 MHz	13.5 MHz	17.7 MHz	14.3 MHz

4.5 Synchronizing Television

Synchronization involves two broad issues:

- Frequency lock
- Phase lock

In the television video context, frequency refers to the number of times a field or frame is recreated over an interval of time. It’s important that the number of frames transmitted over the period of a day, for example, be neither too many or too few. Frequency lock involves keeping the frame count correct over an interval of time.

Phase lock involves keeping multiple video data streams synchronized down to the level of matching individual pixels and it is a much more demanding requirement than frequency lock.

4.6 Computer video

4.6.1 Video concepts

4.6.1.1 Color space

Most computers use the RGB color space (either 15-, 16-, or 24-bit RGB) or 8-bit color index.

4.6.1.2 Square pixels

Most displays for computers have pixels with a 1:1 aspect ratio which are identified as square pixels. In contrast, most consumer television displays have rectangular pixels.

5 Class 2 and Class 3 Streams

The following types of Streams are defined:

– *Asynchronous Stream:* Used to carry intermittent, sporadic information between a Stream Producer and a Stream Consumer;

– *Available Bit Rate Stream:* Used to carry frequent or recurring information, at a rate governed by the System Resource Manager, the Stream Producer and, for Class 2, the Stream Consumer, on one or more Exchanges between a Stream Producer and a Stream Consumer;

– *Real–Time Stream:* Used to carry Real–Time information, with guaranteed Quality of Service parameters, on one or more Exchanges between a Stream Producer and a Stream Consumer.

5.1 Stream basics

If Class 2 or 3 is used to carry Streams, then all Class 1, 2 and 3 traffic shall be carried as Streams, with the following exceptions:

– Stream requests sent to the System Resource Manager (see 5.3.6.1);

– Stream notifications, sent by the System Resource Manager (see 5.3.6.2);

– Stream flow control messages sent by the System Resource Manager (see 5.3.3 and 5.3.7).

5.1.1 Arbitrated Loop Stream characteristics

The following restrictions apply to the Arbitrated Loop topology, when used to carry Streams:

a) round trip latency shall not exceed 75 µs;

b) NL_Ports, operating as Stream Consumers, regardless of Stream Producer requirements, shall issue R_RDYs, assuming a 75 µs round trip latency, at a rate able to support reception of:

1) 2 531 Class 2 or 3 frames, with a 512 byte payload, within 13 625 µs, when operating at a data rate of 1,062 5 GBaud;

2) 2 159 Class 2 or 3 frames, with a 512 byte payload, within 5 812,5 µs, when operating at a data rate of 2,125 GBaud;

3) 1 416 Class 2 or 3 frames, with a 512 byte payload, within 1 906 µs, when operating at a data rate of 4,25 GBaud;

c) Stream Producers, located on an Arbitrated Loop, shall support transmission of:

1) 2 531 Class 2 or 3 frames, with a 512 byte payload, within 14 625 µs, when operating at a data rate of 1,062 5 GBaud;

2) 2 159 Class 2 or 3 frames, with a 512 byte payload, within 6 812,5 µs, when operating at a data rate of 2,125 GBaud;

3) 1 416 Class 2 or 3 frames, with a 512 byte payload, within 2 906 µs, when operating at a data rate of 4,25 GBaud;

d) all transmissions for a single Time Slot shall occur in a single loop tenancy.

5.1.2 Fabric characteristics

The following restrictions apply to the Fabric topology, when used to carry either or both Class 2 and 3 Streams:

a) the uncongested latency, for Class 2 or 3 frames, between any pair of N_Ports, shall not exceed 75 µs;

b) the uncongested latency, for Class 2 or 3 frames, between an NL_Port and an N_Port or an N_Port and an NL_Port, shall not exceed 150 µs;

c) the uncongested latency, for Class 2 or 3 frames, between a pair of NL_Ports on different Arbitrated Loops, shall not exceed 225 µs;

d) Fabric Ports shall issue R_RDYs and provide sufficient BB_Credit, assuming a 10 µs round trip link latency (1 km), to support bidirectional transfer of:

1) 2 531 Class 2 or 3 frames, with a 512 byte payload, within 14 625 µs, when operating at a data rate of 1,062 5 GBaud;

2) 2 159 Class 2 or 3 frames, with a 512 byte payload, within 6 812,5 µs, when operating at a data rate of 2,125 GBaud;

3) 1 416 Class 2 or 3 frames, with a 512 byte payload, within 2 906 µs, when operating at a data rate of 4,25 GBaud;

e) the conditions stated in 5.1.2(d) shall remain true regardless of how many frame sources, Stream Producers, constitute the aggregated frame flow;

f) the conditions stated in 5.1.2(d) shall remain true regardless of the mixture of Class 2 and 3 frames;

g) support a Data Field Size of 2 048 or larger;

h) N_Ports, operating as Stream Consumers, regardless of Stream Producer requirements, shall issue R_RDYs and provide BB_Credit, assuming a 10 µs round trip link latency (1 km), to support reception of:

  1) 2 531 Class 2 or 3 frames, with a 512 byte payload, within 13 625 µs, when operating at a data rate of 1,062 5 GBaud;

  2) 2 159 Class 2 or 3 frames, with a 512 byte payload, within 5 812,5 µs, when operating at a data rate of 2,125 GBaud;

  3) 1 416 Class 2 or 3 frames, with a 512 byte payload, within 1 906 µs, when operating at a data rate of 4,25 GBaud;

i) the Fabric shall support Broadcast;

j) the Fabric should support Multicast;

k) the Fabric shall support the following Well-known addresses:

  1) Broadcast Alias_ID, at hex ‘FFFFFF’;

  2) Link Service Facilitator, at hex ‘FFFFFE’;

  3) Fabric Controller, at hex ‘FFFFFD’;

  4) Directory Server, at hex ‘FFFFFC’, for the purpose of Name Service;

  5) Quality of Service Facilitator, at hex ‘FFFFF9’, which in turn shall support the LSRM (see 5.3.2) and RSRM (see 5.3.5) Extended Link Service commands;

  6) Alias Server, at hex ‘FFFFF8’.

5.1.3 Stream Consumer

Stream Consumers shall:

a) never reassign RX_IDs;

b) support a Data Field Size of 2 048 or larger;

c) never return a P_BSY to any Class 2 Data frame;

d) use ACK_0 for Class 2 flow control;

e) support the following Extended Link Service command[1)]:

  1) GSA (see 5.3.1);

f) register the following with the Name Server:

  1) Node Name;

  2) Port Name;

  3) Class of Service;

  4) FC–4 types;

  5) Port type.

Stream Consumers should:

g) support Broadcast;

h) accept enrolment in Multicast groups;

i) support the following Extended Link Service commands[1)]:

  1) Prompt (see 5.3.3);

  2) Selective Prompt (see 5.3.7).

5.1.4 Stream Definer

Stream Definers shall:

a) support the following Extended Link Service commands[1)]:

  1) GSA (see 5.3.1);

  2) LSRM (5.3.2);

  3) Prompt (see 5.3.3);

  4) SCS (see 5.3.6 and 5.3.6.1);

  5) Selective Prompt (see 5.3.7).

b) register the following with the Name Server:

[1)] Support for Extended Link Service commands not defined in this document (clause), such as PLOGI and FLOGI, are outside the scope of this document (clause).

1) Node Name;
2) Port Name;
3) Class of Service;
4) FC–4 types;
5) Port type.

5.1.5 Stream Producer

Stream Producers shall:

a) never reassign OX_IDs;

b) support a Data Field Size of 2 048 or larger;

c) never cause Sequences to straddle Time Slots;

d) support the following Extended Link Service commands[1]:

1) GSA (see 5.3.1);
2) Prompt (see 5.3.3);
3) SCS (see 5.3.6 and 5.3.6.2);
4) Selective Prompt (see 5.3.7).

e) register the following with the Name Server:

1) Node Name;
2) Port Name;
3) Class of Service;
4) FC–4 types;
5) Port type.

f) support use of ACK_0 for Class 2 flow control;

g) initialize the Asynchronous Stream to the Enabled Stream state (see figure 2);

h) initialize all Available Bit Rate Streams to the Nonexisting Stream state (see figure 2);

i) initialize all Real–Time Streams to the Nonexisting Stream state (see figure 2);

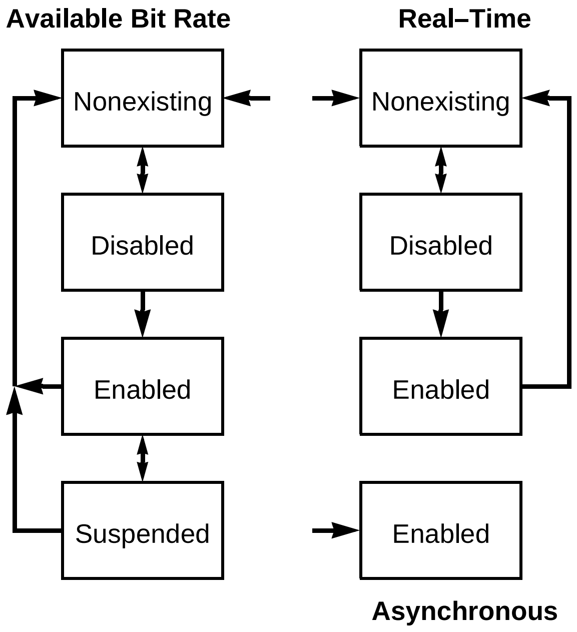

Figure 2 – Stream – States

j) enter the Disabled Stream state, from the Nonexisting Stream state, for an Available Bit Rate Stream, when an accept to a PRABRS or RABRS Stream request is received (see 5.3.6.1);

k) enter the Disabled Stream state, from the Nonexisting Stream state, for an Available Bit Rate Stream, when an ABRSN Stream notification is received (see 5.3.6.2.1);

l) enter the Disabled Stream state, from the Nonexisting Stream state, for a Real–Time Stream, when an accept to a PRRTS or RRTS Stream request is received (see 5.3.6.1);

m) enter the Disabled Stream state, from the Nonexisting Stream state, for a Real–Time Stream Stream, when a RTSN Stream notification is received (see 5.3.6.2.2);

n) enter the Enabled Stream state, from the Disabled Stream state, for an Available Bit Rate Stream, when its Time Slot is active and Mod 2^{32}

$$0 \leq \text{Time Slot Count} - \text{Start Time Slot} \leq 15\ ;$$

o) enter the Enabled Stream state, from the Disabled Stream state, for a Real–Time Stream, when its Time Slot is active and Mod 2^{32}

$$0 \leq \text{Time Slot Count} - \text{Start Time Slot} \leq 15\ ;$$

p) enter the Suspended Stream state, from the Enabled Stream state, for a Real–Time

Stream, when Mod 2^{32}
$0 \le \text{Time Slot Count} - \text{Suspend Time Slot} \le 15$;

q) re-enter the Enabled Stream state, from the Suspended Stream state, for a Real–Time Stream, when Mod 2^{32}
$0 \le \text{Time Slot Count} - \text{ResumeTime Slot} \le 15$;

r) enter the Nonexisting Stream state for an Available Bit Rate Stream, when Mod 2^{32}
$0 \le \text{Time Slot Count} - \text{TerminationTime Slot} \le 2^{31}$
;

s) enter the Nonexisting Stream state for a Real–Time Stream, when Mod 2^{32}
$0 \le \text{Time Slot Count} - \text{TerminationTime Slot} \le 2^{31}$
;

t) only transmit when:

  1) a Stream is in the Enabled state, and

  2) its Time Slot Counter is in the active state, and

  3) the Stream Producer has something to send for the identified Stream.

u) transmit combined Asynchronous Streams, in the order of the Stream replies[2] ;

v) commence transmission of Data frames for Real–Time Stream within a default time of 2 ms, of the reception of the Prompt or Selective Prompt Extended Link Service command, marking the Stream as active;

w) cease transmission of Data frames for Streams, before the end of the Time Slot;

x) support reception of Broadcast frames;

Stream Producers shall, for each Available Bit Rate Stream supported:

y) have one Start Timer;

z) have one Stop Timer, set to a default value of $= \text{StartTime Slot} + 2^{31} - 1$.

Stream Producers Shall, for each Real–Time Stream supported:

aa) have one Start Timer;

ab) have one Pause Timer, set to the default value of $= \text{StartTime Slot} + 2^{31} - 1$;

ac) have one Resume Timer set to the default value of $= \text{StartTime Slot} + 2^{31} - 1$;

ad) have one Stop Timer, set to a default value of $= \text{StartTime Slot} + 2^{31} - 2$.

Stream Producers should:

ae) support reception of Multicast frames;

af) accept enrolment in Multicast groups, setup by the System Resource Manager.

5.1.6 System Resource Manager

System Resource Managers shall:

a) support the following Extended Link Service commands[1]:

  1) GSA (see 5.3.1);

  2) LSRM (see 5.3.2);

  3) Prompt (see 5.3.3);

  4) QSRM (see 5.3.4);

  5) RSRM (see 5.3.5);

  6) SCS (see 5.3.6 and 5.3.6.2).

b) register the following with the Name Server:

  1) Node Name;

  2) Port Name;

  3) Class of Service;

  4) FC–4 types;

  5) Port type.

c) reply to Stream requests for Available Bit Rate and Real–Time Streams within 1/16 second;

d) reply to Stream requests for Asynchronous Streams within 1 second.

e) when combining Asynchronous Stream requests assume:

---

[2] As an example, 4 separate RAS Stream requests have been sent, in the order A, B,C and D, the accept replies are returned D, A, B for Asynchronous Time Slot n. The Stream Producer shall, when it is time to transmit the information for Asynchronous Time Slot n, send the information from request D first, followed by A and B respectively. Request C is assigned to a different Asynchronous Time Slot and are thus not a factor in the transmission for Stream request A, B and D.

1) that each Sequence is transmitted on a separate Exchange;

2) that Concurrent Sequences are set = 1;

3) that each Data frame carry a payload of 2 048 byte;

4) the Stream Producer will not start transmission of a new Sequence until transmission of the current Sequence has completed;

5) combined Stream requests are serviced, by the Stream Producer, in the order of the Stream replies.

f) not cause Asynchronous Stream request for a given Stream Producer, Stream Consumer pair to be serviced in an order different from the order of the Stream requests;

g) maintain the Time Slot interval, with a tolerance of 100 ppm, at:

1) 1/64 second, for 1,062 5 GBaud;

2) 1/128 second, for 2,125 GBaud;

3) 1/256 second, for 4,25 GBaud;

h) increment the Time Slot Number, once, every time a new Time Slot appear;

i) increment each Time Slot Counter, once, every time an active Time Slot occur for the Stream;

j) transmit a Prompt or Selective Prompt Extended Link Service command every time an active Time Slot appear (see 5.3.3 and 5.3.7);

k) transmit a Prompt Extended Link Service command at least once every second (see 5.3.3);

l) base the Stream Time on the common start time of 1 January 1900 (see 5.3.3, 5.3.7 and FC–PH).

System Resource Managers should:

m) combine multiple Asynchronous Stream requests, for service in a single Time Slot.

n) support the Selective Prompt Extended Link Service command (see 5.3.7);

o) establish Multicast groups for transmission of Selective Prompt Extended Link Service commands.

5.2 Predefined Quality of Service Parameters

The following Real–Time Streams, with Predefined Quality of Service Parameters are defined (7 bit):

Table 2 – Predefined Stream

	code hex
Reserved	00
270 Mb/s (601 type) (see 5.2.1)	01
360 Mb/s (601 type NTSC) (see 5.2.2)	02
360 Mb/s (601 type PAL) (see 5.2.3)	03
Reserved	04–7F

5.2.1 Real–Time Stream – 270 Mb/s

This predefined Quality of Service option supports Real–Time transfer of both 601 type PAL and NTSC Streams at 270 Mb/s.

Thirty Time Slots are provided each second, PAL Streams will therefore generally leave 5[3] unused Time Slots per second, which may be used to carry other traffic between the Stream Producer and the Stream Consumer. Transmission of the 270 Mb/s PAL Streams shall commence after at most 1,5 ms, rather than the default 2 ms limit (see 5.1.5(v)).

Table 3 shows the 30 Time Slots assigned to carry the 270 Mb/s Stream.

5.2.2 Real–Time Stream – 360 Mb/s NTSC

This predefined Quality of Service option support of Real–Time transfer of 601 type NTSC Streams at 360 Mb/s.

---

[3] their will occasionally be either 4 or 6 unused Time Slots per second, depending on the clock differences between the 601 source and the System Resource Managers clock. An occasional unused Time Slot will also occur for the NTSC Stream.

Table 3 – 270 Mb/s – 601 type Stream

Data Rate MBaud	Time Slot Number	
1 062,5	1, 3, 5, 7, 9, 11, 13, 15, 18, 20, 22, 24, 26, 28, 30, 33, 35, 37, 39, 41, 43, 45, 47, 50, 52, 54, 56, 58, 60, 62	Mod 64
	2, 4, 6, 8, 10, 12, 14, 17, 19, 21, 23, 25, 27, 29, 31, 34, 36, 38, 40, 42, 44, 46, 49, 51, 53, 55, 57, 59, 61, 63	

Table 4 shows the 30 Time Slot pairs, assigned to carry the 360 Mb/s Stream. The Slot pairs shall for the following purposes be treated as a single Time Slot:

a) termination of transmission, by the Stream Producer (see 5.1.5(c));

b) termination of Loop tenancy (see 5.1.1(d));

c) incrementing of the Time Slot Counter for the Stream (see 5.1.6(i)).

Table 4 – 360 Mb/s – 601 type NTSC

Data Rate MBaud	Time Slot Number	
1 062,5	[1–2], [3–4], [5–6], [7–8], [9–10], [11–12], [13–14], [16–17], [18–19], [20–21], [22–23], [24–25], [26–27], [28–29], [30–31], [33–34], [35–36], [37–38], [39–40], [41–42], [43–44], [45–46], [48–49], [50–51], [52–53], [54–55], [56–57], [58–59], [60–61], [62–63],	Mod 64

5.2.3 Real–Time Stream – 360 Mb/s PAL

This predefined Quality of Service option support of Real–Time transfer of 601 type PAL Streams at 360 Mb/s.

Table 5 shows the 26 Time Slot pairs, assigned to carry the 360 Mb/s Stream. The Slot pairs shall for the following purposes be treated as a single Time Slot:

a) termination of transmission, by the Stream Producer (see 5.1.5(c));

b) termination of Loop tenancy (see 5.1.1(d));

c) incrementing of the Time Slot Counter for the Stream (see 5.1.6(i)).

5.3 Extended Link Service – Stream support

Table 5 – 360 Mb/s – 601 type PAL

Data Rate MBaud	Time Slot Number	
1 062,5	[1–2], [3–4], [6–7], [8–9], [11–12], [13–14], [16–17], [18–19], [20–21], [23–24], [25–26], [28–29], [30–31], [33–34], [35–36], [38–39], [40–41], [43–44], [45–46], [48–49], [50–51], [52–53], [55–56], [57–58], [60–61], [62–63],	Mod 64

The following Extended Link Service commands are defined for the support of Class 2 and 3 Streams:

– *Get Stream Ability (GSA):* Enable a port to obtain the Stream capabilities, parameters of another port;

– *Locate System Resource Manager (LSRM):* Enable an Nx_Port to find the address identifier of the System Resource Manager;

– *Prompt:* Enables the System Resource Manger to control all the Stream flows with a single command;

– *Register System Resource Manager (RSRM):* Enable an Nx_Port to become the System Resource Manager;

– *Query System Resource Manager (QSRM):* Enables discovery of the Available Bit Rate and Real–Time Streams assigned a specific Stream Identifier;

- *Stream Control Services (SCS):* Enables a Stream Definer to establish, suspend, resume or terminate Streams, and the System Resource Manager to notify the Stream Producer of third party establishment, suspension, resumption or termination of Streams;
- *Selective Prompt:* Enables the System Resource Manger to control Stream flows individually.

Support for the above listed Extended Link Service commands shall not require use of PLOGI prior to their use.

5.3.1 Get Stream Ability

Ports send a Get Stream Ability (GSA) Extended Link Service command to a port whose Stream parameters are desired.

The System Resource Manager shall, except as noted below, obtain the Stream parameters of both the Stream Producer and the Stream Consumer before accepting any Stream requests for the Stream Producer and the Stream Consumer.

Stream Definers should, except as noted below, obtain the Stream parameters of both the Stream Producer and the Stream Consumer before issuing any Stream requests for the Stream Producer and the Stream Consumer.

Stream Producers shall obtain the Stream Consumers Stream parameters before accepting establishment of any Stream between the Stream Producer and the Stream Consumer.

Transmission of the GSA and the corresponding Accept shall be sent as an Asynchronous Stream in an individual Asynchronous Stream request (see 5.3.6.1.4) and is not subject to the above stated restrictions.

Protocol:

Get Stream Ability request Sequence
Accept reply Sequence

Format: FT_1

Addressing: The S_ID field designates the source wishing to obtain the destination’s Stream parameters. The D_ID designates the port whose Stream parameters are desired.

Payload: The format of the GSA Extended Link Service command is shown in table 6.

Table 6 – GSA payload

Item	Size (Bytes)
hex ‘xx 00 00 00’	4

Reply Extended Link Service Sequence:

Service Reject (LS_RJT)
Signifies the rejection of the GSA command
Accept (ACC)
Signifies acceptance of the GSA request

- Accept Payload

The format of the Accept Payload is shown in table 7.

Max. number of Sequences for Asynchronous Streams: This 16 bit unsigned field, Async Seq_{max}, shall indicate the maximum number of Sequences, that the sending port can accept in a single Asynchronous Time Slot.

The following relationships shall hold true:

Async $\mathrm{Seq}_{max} \geq$ Class 2 Async Seq_{max}

Async $\mathrm{Seq}_{max} \geq$ Class 3 Async Seq_{max}

Max. number of Data frames for Asynchronous Streams: This 16 bit unsigned field, Async Frame_{max}, shall indicate the maximum number of Data frames, that the sending port can accept in a single Asynchronous Time Slot.

The following relationships shall hold true:

Async $\mathrm{Frame}_{max} \geq$ Class 2 Async Frame_{max}

Async $\mathrm{Frame}_{max} \geq$ Class 3 Async Frame_{max}

Max. number of Class 2 Sequences for Asynchronous Streams: This 16 bit unsigned field, Class 2 Async Seq_{max}, shall indicate the maximum number of Class 2 Sequences, that the sending port can accept in a single Asynchronous Time Slot.

Max. number of Class 2 Data frames for Asynchronous Streams: This 16 bit unsigned field, Class 2 Async Frame_{max}, shall indicate the maximum number of Class 2 Data frames, that the sending port can accept in a single Asynchronous Time Slot.

Table 7 – GSA Accept Payload

Item	Size (Bytes)
hex ‘02 00 00 00’	4
Max. number of Sequences for Asynchronous Streams	2
Max. number of Data frames for Asynchronous Streams	2
Reserved	8
Max. number of Class 2 Sequences for Asynchronous Streams	2
Max. number of Class 2 Data frames for Asynchronous Streams	2
Reserved	2
Max. RX_ID assignment time for Class 2 Asynchronous Stream	2
Max. latency for Class 2 initial ACK	2
Max. latency for Class 2 final ACK	2
Max. number of Class 3 Sequences for Asynchronous Streams	2
Max. number of Class 3 Data frames for Asynchronous Streams	2
SP max. ABR Streams	1
SP max. RT Streams	1
SP max. Streams	1
SP max. aggregate TSs per Stream	1
Reserved	8
SP max. Class 2 ABR Streams	1
SP max. Class 2 RT Streams	1
SP max. Class 2 TSs per RT Stream	1
SP max. aggregate Class 2 TSs per RT Stream	1
Reserved	4
SP max. Class 3 ABR Streams	1
SP max. Class 3 RT Streams	1
SP max. Class 3 TSs per RT Stream	1
SP max. aggregate Class 3 TSs per RT Stream	1
Reserved	4
SC max. ABR Streams	1
SC max. RT Streams	1
SC max. Streams	1
SC max. aggregate TSs per Stream	1
Reserved	20
SC max. Class 2 ABR Streams	1
SC max. Class 2 RT Streams	1
SC max. Class 2 TSs per RT Stream	1
SC max. aggregate Class 2 TSs per RT Stream	1
Reserved	2
SC max. Class 2 Sequences per ABR Stream TS	1
SC max. Class 2 Sequences per RT Stream TS	1
Reserved	2
Max. RX_ID assignment time for Class 2 RT Stream	2
Max. latency for Class 2 initial RT ACK	2
Max. latency for Class 2 final RT ACK	2
Reserved	2
Max. RX_ID assignment time for Class 2 ABR Stream	2
Max. latency for Class 2 initial ABR ACK	2
Max. latency for Class 2 final ABR ACK	2
SC max. Class 3 ABR Streams	1
SC max. Class 3 RT Streams	1
SC max. Class 3 TSs per RT Stream	1
SC max. aggregate Class 3 TSs per RT Stream	1
Reserved	2
SC max. Class 3 Sequences per ABR Stream TS	1
SC max. Class 3 Sequences per RT Stream TS	1

(continued)

Table 7 (concluded)

Max. RX_ID assignment time for Class 2 Asynchronous Stream: This 16 bit unsigned field shall indicate, in units of 1/65 536 of a second, the worst case latency for assignment of an RX_ID to a Sequence received in an Asynchronous Stream.

Max. latency for Class 2 initial ACK: This 16 bit unsigned field shall indicate, in units of 1/65 536 of a second, the worst case latency before the first ACK or P_RJT for a Class 2 Sequence received in an Asynchronous Stream is transmitted.

Max. latency for Class 2 final ACK: This 16 bit unsigned field shall indicate, in units of 1/65 536 of a second, the worst case latency before the final ACK or P_RJT for a Class 2 Sequence received in an Asynchronous Stream is transmitted.

Max. number of Class 3 Sequences for Asynchronous Streams: This 16 bit unsigned field, $\text{Class 3 Async Seq}_{\text{max}}$, shall indicate the maximum number of Class 3 Sequences, that the sending port can accept in a single Asynchronous Time Slot.

Max. number of Class 3 Data frames for Asynchronous Streams: This 16 bit unsigned field, $\text{Class 3 Async Frame}_{\text{max}}$, shall indicate the maximum number of Class 3 Data frames, that the sending port can accept in a single Asynchronous Time Slot.

SP max. ABR Streams: This 8 bit unsigned field, $\text{SP ABR Stream}_{\text{max}}$, shall indicate the maximum number of Available Bit Rate Streams the port is able to support as a Stream Producer.

The following relationships shall hold true:

$\text{SP ABR Stream}_{\text{max}} \geq \text{Class 2 SP ABR Stream}_{\text{max}}$

$\text{SP ABR Stream}_{\text{max}} \geq \text{Class 3 SP ABR Stream}_{\text{max}}$

SP max. RT Streams: This 8 bit unsigned field, $\text{SP RT Stream}_{\text{max}}$, shall indicate the maximum number of Real–Time Streams the port is able to support as a Stream Producer.

The following relationships shall hold true:

$\text{SP RT Stream}_{\text{max}} \geq \text{Class 2 SP RT Stream}_{\text{max}}$

$\text{SP RT Stream}_{\text{max}} \geq \text{Class 3 SP RT Stream}_{\text{max}}$

SP max. Streams: This 8 bit unsigned field, $\text{SP Stream}_{\text{max}}$, shall indicate the maximum number of Available Bit Rate and Real–Time Streams the port is able to support as a Stream Producer.

The following relationships shall hold true:

$\text{SP Stream}_{\text{max}} \geq \text{SP ABR Stream}_{\text{max}}$

$\text{SP Stream}_{\text{max}} \geq \text{SP RT Stream}_{\text{max}}$

SP max. aggregate TSs per Stream: This 8 bit unsigned field, $\text{SP TS agg}_{\text{max}}$, shall indicate the maximum number of Time Slots per second, assigned to Available Bit Rate and Real–Time Streams, that the port is able to support as a Stream Producer.

The following relationships shall hold true:

$\text{SP TS agg}_{\text{max}} \geq \text{Class 2 SP RT TS agg}_{\text{max}}$

$\text{SP TS agg}_{\text{max}} \geq \text{Class 3 SP RT TS agg}_{\text{max}}$

SP max. Class 2 ABR Streams: This 8 bit unsigned field shall indicate the maximum number of Class 2 Available Bit Rate Streams the port is able to support as a Stream Producer.

SP max. Class 2 RT Streams: This 8 bit unsigned field, $\text{Class 2 SP RT Stream}_{\text{max}}$, shall indicate the maximum number of Class 2 Real–Time Streams the port is able to support as a Stream Producer.

SP max. Class 2 TSs per RT Stream: This 8 bit unsigned field, $\text{Class 2 SP RT TS}_{\text{max}}$, shall indicate, as a Stream Producer, the maximum number of Time Slots per second, which may be assigned to a single Class 2 Real–Time Stream.

SP max. aggregate Class 2 TSs per RT Stream: This 8 bit unsigned field, $\text{Class 2 SP RT TS agg}_{\text{max}}$, shall indicate the maximum number of Time Slots per second, assigned to Class 2 Real–Time Streams, that the port is able to support as a Stream Producer.

The following relationship shall hold true:

$\text{Class 2 SP RT TS agg}_{\text{max}} \geq \text{Class 2 SP RT TS}_{\text{max}}$

SP max. Class 3 ABR Streams: This 8 bit unsigned field, $\text{Class 3 SP ABR Stream}_{\text{max}}$, shall indicate the maximum number of Class 3 Available Bit Rate Streams the port is able to support as a Stream Producer.

SP max. Class 3 RT Streams: This 8 bit unsigned field, $\text{Class 3 SP RT Stream}_{max}$, shall indicate the maximum number of Class 3 Real–Time Streams the port is able to support as a Stream Producer.

SP max. Class 3 TSs per RT Stream: This 8 bit unsigned field, $\text{Class 3 SP RT TS}_{max}$, shall indicate the maximum number of Time Slots per second, which may be assigned to a single Class 3 Real–Time Stream.

SP max. aggregate Class 3 TSs per RT Stream: This 8 bit unsigned field, $\text{Class 3 SP RT TS agg}_{max}$, shall indicate the maximum number of Time Slots per second, assigned to Class 3 Real–Time Streams, that the port is able to support as a Stream Producer.

The following relationship shall hold true:

$\text{Class 3 SP RT TS agg}_{max} \geq \text{Class 3 SP RT TS}_{max}$

SC max. ABR Streams: This 8 bit unsigned field, $\text{SC ABR Stream}_{max}$, shall indicate the maximum number of Available Bit Rate Streams the port is able to support as a Stream Consumer.

The following relationships shall hold true:

$\text{SC ABR Stream}_{max} \geq \text{Class 2 SC ABR Stream}_{max}$

$\text{SC ABR Stream}_{max} \geq \text{Class 3 SC ABR Stream}_{max}$

SC max. RT Streams: This 8 bit unsigned field, $\text{SC RT Stream}_{max}$, shall indicate the maximum number of Real–Time Streams the port is able to support as a Stream Consumer.

The following relationships shall hold true:

$\text{SC RT Stream}_{max} \geq \text{Class 2 SC RT Stream}_{max}$

$\text{SC RT Stream}_{max} \geq \text{Class 3 SC RT Stream}_{max}$

SC max. Streams: This 8 bit unsigned field, SC Stream_{max}, shall indicate the maximum number of Available Bit Rate and Real–Time Streams the port is able to support as a Stream Consumer.

The following relationships shall hold true:

$\text{SC Stream}_{max} \geq \text{SC ABR Stream}_{max}$

$\text{SC Stream}_{max} \geq \text{SC RT Stream}_{max}$

SC max. aggregate TSs per Stream: This 8 bit unsigned field, SC TS agg_{max}, shall indicate the maximum number of Time Slots per second, assigned to Available Bit Rate and Real–Time Streams, that the port is able to support as a Stream Consumer.

The following relationships shall hold true:

$\text{SC TS agg}_{max} \geq \text{Class 2 SC RT TS agg}_{max}$

$\text{SC TS agg}_{max} \geq \text{Class 3 SC RT TS agg}_{max}$

SC max. Class 2 ABR Streams: This 8 bit unsigned field, $\text{Class 2 SC ABR Stream}_{max}$, shall indicate the maximum number of Class 2 Available Bit Rate Streams the port is able to support as a Stream Consumer.

SC max. Class 2 RT Streams: This 8 bit unsigned field, $\text{Class 2 SC RT Stream}_{max}$, shall indicate the maximum number of Class 2 Real–Time Streams the port is able to support as a Stream Consumer.

SC max. Class 2 TSs per RT Stream: This 8 bit unsigned field, $\text{Class 2 SC RT TS}_{max}$, shall indicate, as a Stream Consumer, the maximum number of Time Slots per second, which may be assigned to a single Class 2 Real–Time Stream.

SC max. aggregate Class 2 TSs per RT Stream: This 8 bit unsigned field, $\text{Class 2 SC RT TS agg}_{max}$, shall indicate the maximum number of Time Slots per second, assigned to Class 2 Real–Time Streams, that the port is able to support as a Stream Consumer.

The following relationship shall hold true:

$\text{Class 2 SC RT TS agg}_{max} \geq \text{Class 2 SC RT TS}_{max}$

SC max. Class 2 Sequences per ABR Stream TS: This 8 bit unsigned field shall indicate the maximum number of Class 2 Sequences, that the sending port can accept in a single Time Slot, used to carry an Available Bit Rate Stream.

SC max. Class 2 Sequences per RT Stream TS: This 8 bit unsigned field shall indicate the maximum number of Class 2 Sequences, that the sending port can accept in a single Time Slot, used to carry a Real–Time Stream.

Max. RX_ID assignment time for Class 2 RT Stream: This 16 bit unsigned field shall indicate, in units of 1/65 536 of a second, the worst case latency for assignment of an RX_ID to a Sequence received in a Real–Time Stream.

Max. latency for Class 2 initial RT ACK: This 16 bit unsigned field shall indicate, in units of 1/65 536 of a second, the worst case latency before the first ACK or P_RJT for a Class 2 Sequence received in a Real–Time Stream is transmitted.

Max. latency for Class 2 final RT ACK: This 16 bit unsigned field shall indicate, in units of 1/65 536 of a second, the worst case latency before the final ACK or P_RJT for a Class 2 Sequence received in a Real–Time Stream is transmitted.

Max. RX_ID assignment time for Class 2 ABR Stream: This 16 bit unsigned field shall indicate, in units of 1/65 536 of a second, the worst case latency for assignment of an RX_ID to a Sequence received in an Available Bit Rate Stream.

Max. latency for Class 2 initial ABR ACK:

This 16 bit unsigned field shall indicate, in units of 1/65 536 of a second, the worst case latency before the first ACK or P_RJT for a Class 2 Sequence received in an Available Bit Rate Stream is transmitted.

Max. latency for Class 2 final ABR ACK:

This 16 bit unsigned field shall indicate, in units of 1/65 536 of a second, the worst case latency before the final ACK or P_RJT for a Class 2 Sequence received in an Available Bit Rate Stream is transmitted.

SC max. Class 3 ABR Streams: This 8 bit unsigned field, $\text{Class 3 SC ABR Stream}_{\text{max}}$, shall indicate the maximum number of Class 3 Available Bit Rate Streams the port is able to support as a Stream Consumer.

SC max. Class 3 RT Streams: This 8 bit unsigned field, $\text{Class 3 SC RT Stream}_{\text{max}}$, shall indicate the maximum number of Class 3 Real–Time Streams the port is able to support as a Stream Consumer.

SC max. Class 3 TSs per RT Stream: This 8 bit unsigned field, $\text{Class 3 SC RT TS}_{\text{max}}$, shall indicate the maximum number of Time Slots per second, which may be assigned to a single Class 3 Real–Time Stream.

SC max. aggregate Class 3 TSs per RT Stream: This 8 bit unsigned field, $\text{Class 3 SC RT TS agg}_{\text{max}}$, shall indicate the maximum number of Time Slots per second, assigned to Class 3 Real–Time Streams, that the port is able to support as a Stream Consumer.

The following relationship shall hold true:

$\text{Class 3 SC RT TS agg}_{\text{max}} \geq \text{Class 3 SC RT TS}_{\text{max}}$

SC max. Class 3 Sequences per ABR Stream TS: This 8 bit unsigned field shall indicate the maximum number of Class 3 Sequences, that the sending port can accept in a single Time Slot, used to carry an Available Bit Rate Stream.

SC max. Class 3 Sequences per RT Stream TS: This 8 bit unsigned field shall indicate the maximum number of Class 3 Sequences, that the sending port can accept in a single Time Slot, used to carry a Real–Time Stream.

5.3.2 System Resource Manager – Locate

Ports send a Locate System Resource Manager (LSRM) Extended Link Service command to the Quality of Service Facilitator (QoSF), located at the well-known address hex ‘FFFFF9’, to find the Address Identifier behind which the System Resource Manager is located.

Protocol:

Locate System Resource Manager request Sequence
Accept reply Sequence

Format: FT_1

Addressing: The S_ID field designates the source Port requesting the Address Identifier, pinpointing the location of the System Resource Manager. The D_ID designates the Quality of Service Facilitator, hex ‘FFFFF9’.

The LSRM request shall be rejected (LS_RJT) with a reason code explanation of ‘No System

Resource Manager registered' if the Quality of Service Facilitator is unaware of the presence of a System Resource Manager.

Payload: The format of the LSRM Extended Link Service command is shown in table 8.

Table 8 – LSRM payload

Item	Size (Bytes)
hex 'xx 00 00 00'	4

Reply Extended Link Service Sequence:

Service Reject (LS_RJT)

Signifies the rejection of the LSRM command

Accept (ACC)

Signifies acceptance of the LSRM request

– Accept Payload

The format of the Accept Payload is shown in table 9.

Table 9 – LSRM Accept Payload

Item	Size (Bytes)
hex '02 00 00 00'	4
Reserved	1
Native Port Identifier of the System Resource Manager	3

Native Port Identifier of the System Resource Manager: The Address Identifier used by the System Resource Manager, to accept Stream Control Service (SCS) requests and transmit Stream notifications.

5.3.3 System Resource Manager – Prompt

The System Resource Manager broadcasts a Prompt Extended Link Service command, to mark the beginning of a new Time Slot. The Prompt Extended Link Service command includes the current Time Slot Count for each of the 256 Streams, along with an indication of which Streams are permitted to use the new Time Slot.

Protocol:

Prompt request Sequence (Single frame)

Format: FT_1

Addressing: The D_ID field designates the well–known Broadcast address hex 'FFFFFF'. The S_ID field designates the System Resource Manager.

Payload: The format of the Prompt Extended Link Service command is shown in table 10.

Table 10 – Prompt payload

Item	Size (Bytes)
hex 'xx 00 00 00'	4
Time Slot Number	4
Stream Time	8
Active Time Slot	32
Time Slot Counter for Stream # 0	4
...	
Time Slot Counter for Stream # 255	4

Time Slot Number: An unsigned 32 bit field indicating the current Time Slot Number. Time Slots are numbered consecutively, wrapping from hex 'FFFFFFFF' to hex '00000000'. The Time Slot Number serves to date SCS requests and replies.

Stream Time: A 64 bit time value, using the format specified by FC–PH for the Expiration timer, providing the System Stream Time, for the use of Stream Definers and Stream Producers.

Active Time Slot: The Active Time Slot field identifies the Streams that have permission to transmit in the current Time Slot. The format of the Active Time Slot field shall be bit mapped as shown below:

- the 3 most significant bits of the Stream Identifier shall be used to identify the word for the Active Time Slot field;
  - Word 0 contains information related to Stream Identifier hex '00' through hex '1F';

Table 11 – Active Time Slots – mapping

Stream Identifier Bit 4 3210	Stream Identifier							
	Bit 7 6 5	Bit 7 6 5	Bit 7 6 5	Bit 7 6 5	Bit 7 6 5	Bit 7 6 5	Bit 7 6 5	Bit 7 6 5
	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0 0000	wd 0 [0]	wd 1 [0]	wd 2 [0]	wd 3 [0]	wd 4 [0]	wd 5 [0]	wd 6 [0]	wd 7 [0]
0 0001	wd 0 [1]	wd 1 [1]	wd 2 [1]	wd 3 [1]	wd 4 [1]	wd 5 [1]	wd 6 [1]	wd 7 [1]
0 0010	wd 0 [2]	wd 1 [2]	wd 2 [2]	wd 3 [2]	wd 4 [2]	wd 5 [2]	wd 6 [2]	wd 7 [2]
0 0011	wd 0 [3]	wd 1 [3]	wd 2 [3]	wd 3 [3]	wd 4 [3]	wd 5 [3]	wd 6 [3]	wd 7 [3]
0 0100	wd 0 [4]	wd 1 [4]	wd 2 [4]	wd 3 [4]	wd 4 [4]	wd 5 [4]	wd 6 [4]	wd 7 [4]
0 0101	wd 0 [5]	wd 1 [5]	wd 2 [5]	wd 3 [5]	wd 4 [5]	wd 5 [5]	wd 6 [5]	wd 7 [5]
0 0110	wd 0 [6]	wd 1 [6]	wd 2 [6]	wd 3 [6]	wd 4 [6]	wd 5 [6]	wd 6 [6]	wd 7 [6]
0 0111	wd 0 [7]	wd 1 [7]	wd 2 [7]	wd 3 [7]	wd 4 [7]	wd 5 [7]	wd 6 [7]	wd 7 [7]
0 1000	wd 0 [8]	wd 1 [8]	wd 2 [8]	wd 3 [8]	wd 4 [8]	wd 5 [8]	wd 6 [8]	wd 7 [8]
0 1001	wd 0 [9]	wd 1 [9]	wd 2 [9]	wd 3 [9]	wd 4 [9]	wd 5 [9]	wd 6 [9]	wd 7 [9]
0 1010	wd 0 [10]	wd 1 [10]	wd 2 [10]	wd 3 [10]	wd 4 [10]	wd 5 [10]	wd 6 [10]	wd 7 [10]
0 1011	wd 0 [11]	wd 1 [11]	wd 2 [11]	wd 3 [11]	wd 4 [11]	wd 5 [11]	wd 6 [11]	wd 7 [11]
0 1100	wd 0 [12]	wd 1 [12]	wd 2 [12]	wd 3 [12]	wd 4 [12]	wd 5 [12]	wd 6 [12]	wd 7 [12]
0 1101	wd 0 [13]	wd 1 [13]	wd 2 [13]	wd 3 [13]	wd 4 [13]	wd 5 [13]	wd 6 [13]	wd 7 [13]
0 1110	wd 0 [14]	wd 1 [14]	wd 2 [14]	wd 3 [14]	wd 4 [14]	wd 5 [14]	wd 6 [14]	wd 7 [14]
0 1111	wd 0 [14]	wd 1 [14]	wd 2 [14]	wd 3 [14]	wd 4 [14]	wd 5 [14]	wd 6 [14]	wd 7 [14]
1 0000	wd 0 [16]	wd 1 [16]	wd 2 [16]	wd 3 [16]	wd 4 [16]	wd 5 [16]	wd 6 [16]	wd 7 [16]
1 0001	wd 0 [17]	wd 1 [17]	wd 2 [17]	wd 3 [17]	wd 4 [17]	wd 5 [17]	wd 6 [17]	wd 7 [17]
1 0010	wd 0 [18]	wd 1 [18]	wd 2 [18]	wd 3 [18]	wd 4 [18]	wd 5 [18]	wd 6 [18]	wd 7 [18]
1 0011	wd 0 [19]	wd 1 [19]	wd 2 [19]	wd 3 [19]	wd 4 [19]	wd 5 [19]	wd 6 [19]	wd 7 [19]
1 0100	wd 0 [20]	wd 1 [20]	wd 2 [20]	wd 3 [20]	wd 4 [20]	wd 5 [20]	wd 6 [20]	wd 7 [20]
1 0101	wd 0 [21]	wd 1 [21]	wd 2 [21]	wd 3 [21]	wd 4 [21]	wd 5 [21]	wd 6 [21]	wd 7 [21]
1 0110	wd 0 [22]	wd 1 [22]	wd 2 [22]	wd 3 [22]	wd 4 [22]	wd 5 [22]	wd 6 [22]	wd 7 [22]
1 0111	wd 0 [23]	wd 1 [23]	wd 2 [23]	wd 3 [23]	wd 4 [23]	wd 5 [23]	wd 6 [23]	wd 7 [23]
1 1000	wd 0 [24]	wd 1 [24]	wd 2 [24]	wd 3 [24]	wd 4 [24]	wd 5 [24]	wd 6 [24]	wd 7 [24]
1 1001	wd 0 [25]	wd 1 [25]	wd 2 [25]	wd 3 [25]	wd 4 [25]	wd 5 [25]	wd 6 [25]	wd 7 [25]
1 1010	wd 0 [26]	wd 1 [26]	wd 2 [26]	wd 3 [26]	wd 4 [26]	wd 5 [26]	wd 6 [26]	wd 7 [26]
1 1011	wd 0 [27]	wd 1 [27]	wd 2 [27]	wd 3 [27]	wd 4 [27]	wd 5 [27]	wd 6 [27]	wd 7 [27]
1 1100	wd 0 [28]	wd 1 [28]	wd 2 [28]	wd 3 [28]	wd 4 [28]	wd 5 [28]	wd 6 [28]	wd 7 [28]
1 1101	wd 0 [29]	wd 1 [29]	wd 2 [29]	wd 3 [29]	wd 4 [29]	wd 5 [29]	wd 6 [29]	wd 7 [29]
1 1110	wd 0 [30]	wd 1 [30]	wd 2 [30]	wd 3 [30]	wd 4 [30]	wd 5 [30]	wd 6 [30]	wd 7 [30]
1 1111	wd 0 [31]	wd 1 [31]	wd 2 [31]	wd 3 [31]	wd 4 [31]	wd 5 [31]	wd 6 [31]	wd 7 [31]

– Word 1 contains information related to Stream Identifier hex ‘20’ through hex ‘3F’;

– Word 2 contains information related to Stream Identifier hex ‘40’ through hex ‘5F’;

– Word 3 contains information related to Stream Identifier hex ‘60’ through hex ‘7F’;

– Word 4 contains information related to Stream Identifier hex ‘80’ through hex ‘9F’;

– Word 5 contains information related to Stream Identifier hex ‘A0’ through hex ‘BF’;

– Word 6 contains information related to Stream Identifier hex ‘C0’ through hex ‘DF’;

– Word 7 contains information related to Stream Identifier hex ‘E0’ through hex ‘FF’.

– the 5 least significant bits of the Stream Identifier shall be used to identify the bit position within the word for the Active Time Slot field (see table 11).

– To mark a specific Time Slot Counter as active, the bit, identified using the above procedure, shall be set = 1, a value of = 0 shall indicate, that the identified Time Slot Counter is inactive.

Time Slot Counter for Stream #n: The Time Slot Counters shall for Stream ‘n’, indicate the current Time Slot Value.

This filed is replicated 256 times, once for every potential Stream Identifier.

Reply Extended Link Service Sequence:

None

5.3.4 System Resource Manager – Query

Ports send a Query System Resource Manager (QSRM) Extended Link Service command to the System Resource Manager to discover which Available Bit Rate or Real–Time Stream uses a specific Stream Identifier.

Protocol:

Query System Resource Manager request Sequence
Accept reply Sequence

Format: FT_1

Addressing: The S_ID field designates the source Port wishing information relating to a specific Stream Identifier. The D_ID designates the System Resource Manager.

Payload: The format of the QSRM Extended Link Service command is shown in table 12.

Table 12 – QSRM payload

Item	Size (Bytes)
hex ‘xx 00 00 00’	4
Stream Identifier	1
Reserved	3

Stream Identifier: An unsigned 8 bit identifier, assigned by the System Resource Manager to a Available Bit Rate or Real–Time Stream. The Stream Identifier shall not be set = 0, which are used for the Asynchronous Streams.

Reply Extended Link Service Sequence:

Service Reject (LS_RJT)
Signifies the rejection of the QSRM command
Accept (ACC)
Signifies acceptance of the QSRM request

– Accept Payload

The format of the Accept Payload is shown in table 13.

Table 13 – QSRM Accept Payload

Item	Size (Bytes)
hex ‘02 00 00 00’	4
Stream Time	8
Response #1	28
…	
Response #n	28

Stream Time: The Stream Time field values shall indicate the System Stream Time at which the response was generated. The format of the 64 bit Stream Time field shall be as specified by FC–PH for the Expiration timer.

Response #m: The format of each response is shown in table 14. If no Available Bit Rate or Real–Time Stream is assigned to the queried Stream Identifier, then no Response field will be returned in the QSRM Accept payload.

Control: An unsigned 8 bit field, consisting of three sub-fields, an End of Response field, a Priority field and a Class of Service field.

The single bit End of Response field is used to mark the last Response in the QSRM accept payload. This field, using bit 31 of the first word in the QSRM response, shall be set = 1 on the last Response and set = 0 on all other Responses.

The 3 bit Priority field is limited to the range of 0, highest priority, to 7, the lowest priority level. The Priority field uses bits 28 through 30 of the first word in the QSRM response.

The third field, using bits 24 through 27 in the first word, indicate the Class of Service which

Table 14 – QSRM – Response

Item	Size (Bytes)
Control	1
Stream Producer	3
Stream Type	1
Stream Consumer	3
Start Time fraction (1/256 s)	1
Stop Time fraction (1/256 s)	1
Pause Time fraction (1/256 s)	1
Resume Time fraction (1/256 s)	1
Start Time unit (s)	4
Stop Time unit (s)	4
Pause Time unit (s)	4
Resume Time unit (s)	4

will be used to transmit the requested Stream. The encoding of the Class of Service field is as shown in table 15.

Table 15 – Class of Service – Encoding

Item	Code
Reserved	0 – 1
Class 2 Service	2
Class 3 Service	3
Reserved	4 – F

Stream Producer: A 24 bit address identifier, indicating the source of the Available Bit Rate or Real–Time Stream.

Stream Type: The Stream Type field shall indicate the type of Stream established between the Stream Producer and the Stream Consumer. The Stream Type field shall be set = 2 to indicate an Available Bit rate Stream. The Stream Type field shall be set = 3 to indicate a Real–Time Stream.

Stream Consumer: A 24 bit address identifier, indicating the destination of the Available Bit Rate or Real–Time Stream.

Start Time fraction: The Start Time fraction field shall, in units of 1/256 s, indicate an offset, delay, from the Start Time unit field, before Stream transfer is to commence.

Stop Time fraction: The Stop Time fraction field shall, in units of 1/256 s, indicate an offset, delay, from the Stop Time unit field, before Stream transfer is to cease.

The Stop Time fraction field shall be reserved, if the Stop Time unit field, indicates an unspecified termination time.

Pause Time fraction: The Pause Time fraction field shall, in units of 1/256 s, indicate an offset, delay, from the Pause Time unit field, before Stream transfer is to be suspended.

The Pause Time fraction field shall be reserved, if the Stream Type is set to Available Bit Rate or if the Pause Time unit field, indicates an unspecified suspension time.

Resume Time fraction: The Resume Time fraction field shall, in units of 1/256 s, indicate an offset, delay, from the Resume Time unit field, before Stream transfer is to resume.

The Resume Time fraction field shall be reserved, if the Stream Type is set to Available Bit Rate or if the Resume Time unit field, indicates an unspecified resumption time.

Start Time unit: The Start Time unit field indicates in System Stream Time seconds, the instant at which Stream transfer is to commence.

Stop Time unit: The Stop Time unit field indicates in System Stream Time seconds, the instant at which Stream transfer is to cease.

The Stop Time unit field shall be set = 0 to indicate an unspecified termination time for the Stream.

Pause Time unit: The Pause Time unit field indicates in System Stream Time seconds, the instant at which Stream transfer is to be suspended.

The Pause Time unit field shall be reserved, if the Stream Type is set to Available Bit Rate.

The Pause Time unit field shall be set = 0 to indicate an unspecified suspension time for the Stream.

Resume Time unit: The Resume Time unit field indicates in System Stream Time seconds, the instant at which Stream transfer is to be resumed.

The Resume Time unit field shall be reserved, if the Stream Type is set to Available Bit Rate.

The Resume Time unit field shall be set = 0 to indicate an unspecified resumption time for the Stream.

5.3.5 System Resource Manager – Register

Ports sends an Register System Resource Manager (RSRM) Extended Link Service command to the Quality of Service Facilitator (QoSF), located at the well-known address hex ‘FFFFF9’, to register their ability to become the System Resource Manager.

Protocol:

Register System Resource Manager request Sequence
Accept reply Sequence

Format: FT_1

Addressing: The S_ID field designates the source Port wishing to become the System Resource Manager. The D_ID designates the Quality of Service Facilitator, hex ‘FFFFF9’.

Payload: The format of the RSRM Extended Link Service command is shown in table 16.

Table 16 – RSRM payload

Item	Size (Bytes)
hex ‘xx 00 00 00’	4
Reserved	1
Native Port Identifier of the proposed System Resource Manager	3

Native Port Identifier of the proposed System Resource Manager: The Address Identifier to be used by the System Resource Manager, to accept Stream Control Service (SCS) requests and transmit Stream notifications.

Reply Extended Link Service Sequence:

Service Reject (LS_RJT)
Signifies the rejection of the RSRM command
Accept (ACC)
Signifies acceptance of the RSRM request

– Accept Payload

The format of the Accept Payload is shown in table 17.

Table 17 – RSRM Accept Payload

Item	Size (Bytes)
hex ‘02 00 00 00’	4

5.3.6 Stream Control Services

The Stream Control Services (SCS) Extended Link Service command shall be used by Stream Definers to request establishment, suspension, resumption and termination of Class 2 or 3 Streams. The SCS Extended Link Service command shall be used by the System Resource Manager to notify Stream Producers of third party establishment, suspension, resumption and termination request of Class 2 or 3 Streams.

Stream Definers may send a single SCS reservation request Sequence every Slot period. Stream Definers shall not send more than one SCS Extended Link Service Command per Time Slot period.

A SCS Sequence may contain multiple Stream requests. The SCS Sequence is limited in size to a maximum of 14 Stream requests. If multiple Stream requests are made in an SCS request Sequence, then they shall be sorted according to Priority order, with higher priority Stream requests appearing before lower order priority Stream requests. Furthermore, a Stream Definer is limited to a maximum of 256 pending Stream requests at any given time.

Protocol:

Register System Resource Manager request Sequence (Single frame)
Accept reply Sequence

Format: FT_1

Addressing: The S_ID field designates the Stream Definer, for Stream requests, or the System Resource Manager, for Stream notifications. The D_ID designates the System Resource Manager, if the S_ID identify a Stream Definer or a Stream Producer, if the S_ID identify the System Resource Manager.

Payload: The format of the SCS Extended Link Service command is shown in table 18.

Table 18 – SCS Payload

Item	Size (Bytes)
hex ‘xx 00 00 00’	4
Time Slot Number	4
SCS request or notification #1	16
…	
SCS request or notification #n	16

Time Slot Number: The Time Slot Number during which the SCS request Sequence was queued for transmission.

SCS request or notification: If the SCS Extended Link Service Sequence is sent by a Stream Definer, then this field shall follow one of the Stream requests defined in 5.3.6.1. If the SCS Extended Link Service Sequence is sent by a System Resource Manager, then this field shall follow one of the Stream notifications defined in 5.3.6.2.

Reply Extended Link Service Sequence:

Service Reject (LS_RJT)
 Signifies the rejection of the SCS command
Accept (ACC)
 Signifies acceptance of the SCS request

– Accept Payload

The format of the Accept Payload is shown in table 19.

Time Slot Number: The Time Slot Number during which the Accept for the SCS request Sequence was queued for transmission.

SCS response: One of the SCS responses, defined in see 5.3.6.3.

Table 19 – SCS Accept Payload

Item	Size (Bytes)
hex ‘02 00 00 00’	4
Time Slot Number	4
SCS response #1	16
…	
SCS response #m	16

NOTE – The number and order of the SCS responses does not match the SCS request number or order, as replies to SCS requests may appear in separate SCS Accept Sequences.

5.3.6.1 Stream Control Service – Requests

The following Stream reservation requests are defined:

– *No Operation (NOP):* Used to pass an empty Stream request to the System Resource Manager;

– *Preemptive Reserve Available Bit Rate Stream (PRABRS):* Used to establish an Available Bit Rate Stream between a Stream Producer and a Stream Consumer, preempting any lower priority Streams;

– *Preemptive Reserve Real–Time Stream (PRRTS):* Used to establish a Real–Time Stream between a Stream Producer and a Stream Consumer, preempting any lower priority Streams;

– *Reserve Asynchronous Stream (RAS):* Used to reserve a single Time Slot for the transfer of a limited number of Data frames between a Stream Producer and a Stream Consumer;

– *Reserve Available Bit Rate Stream (RABRS):* Used to establish an Available Bit Rate Stream between a Stream Producer and a Stream Consumer;

– *Reserve Real–Time Stream (RRTS):* Used to establish a Real–Time Stream between a Stream Producer and a Stream Consumer;

– *Resume Real–Time Stream (RSRTS):* Used to restart a suspended Real–Time Stream.

– *Suspend Real–Time Stream (SRTS):* Used to quiesce a Real–Time Stream;

– *Terminate Available Bit Rate Stream (TABRS):* Used to remove an Available Bit Rate Stream, releasing the resource allocated to the Stream;

– *Terminate Real–Time Stream (TRTS):* Used to remove an Real–Time Stream, releasing the resource allocated to the Stream;

5.3.6.1.1 No Operation

The No Operation (NOP) request is used by a Steam Definer, to pass an empty Stream request to the System Resource Manager.

Payload: The format of the NOP request is shown in table 20.

Table 20 – NOP (00)

Item	Size (Bytes)
Control	1
Stream Command Code (hex '00')	1
Reserved	1
Transaction Identifier	1
Reserved	12

Control: An unsigned 8 bit field, consisting of two 4 bit sub-fields, a Priority field and a Class of Service field. The 4 bit Priority field is limited to the range of 0, highest priority, to 7, the lowest priority level. The Priority field uses bits 28 through 31 of the first word in the SCS request. The second field, the Class of Service field, using bits 24 through 27 in the first word, is reserved, and shall be set = 0.

Stream Command Code: An 8 bit field uniquely identifying the Stream request. The SCC for the NOP Stream request shall be hex '00'.

Transaction Identifier: An 8 bit identifier, assigned by the Stream Definer. The Transaction Identifier ensure that the Stream Definer can match the Accept or Reject Stream reply to the initial NOP Stream request.

Replies to Stream requests:

Stream request rejection
Signifies the rejection of the NOP request

– Reject payload; the format of the Reject response is shown in 5.3.6.3.2.

Stream request acceptance
Signifies acceptance of the NOP request

– Accept payload; the format of the Accept response is shown in 5.3.6.3.1.2.

5.3.6.1.2 Preemptive Reserve Available Bit Rate Stream

The Preemptive Reserve Available Bit Rate Stream (PRABRS) Stream request is used by a Stream Definer to request reservation for the transfer of one or more Sequences, in a variable Available Bit Rate Time Slot pattern, from a Stream Producer, to a given destination, the Stream Consumer, starting at a specific time.

The System Resource Manager will, to successfully honor the Stream request, terminate lower priority Available Bit Rate or Real–Time Streams, blocking the PRABRS.

The Stream Definer, need not be the Stream Producer nor the Stream Consumer.

The Stream Consumer, may for Class 3 Available Bit Rate Streams be a Multicast group or the well-known Broadcast address hex 'FFFFFF'.

Payload: The format of the PRABRS Stream request is shown in table 21.

Control: An unsigned 8 bit field, consisting of two 4 bit sub-fields, a Priority field and a Class of Service field. The 4 bit Priority field is limited to the range of 0, highest priority, to 7, the lowest priority level. The Priority field uses bits 28 through 31 of the first word in the SCS request. The second field, using bits 24 through 27 in the first word, indicate the Class of Service which will be used to transmit the requested Stream. The encoding of the Class of Service field is as shown in table 15.

Stream Command Code: An 8 bit field uniquely identifying the Stream request. The SCC for the PRABRS Stream request shall be hex '06'.

Table 21 – PRABRS (06)

Item	Size (Bytes)
Control	1
Stream Command Code (hex ‘06’)	1
Reserved	1
Transaction Identifier	1
Producer Tag	1
Stream Producer	3
Start Time fraction (1/256 s)	1
Stream Consumer	3
Start Time (s)	4

Transaction Identifier: An 8 bit identifier, assigned by the Stream Definer. The Transaction Identifier ensure that the Stream Definer can match the Accept or Reject Stream reply to the initial PRABRS Stream request.

Producer Tag: An unique 8 bit identifier, assigned by the Stream Producer to a specific Audio Visual piece of material or source, selected by a method outside the scope of this document.

Stream Producer: A 24 bit address identifier, indicating the source of the Available Bit Rate Stream.

Start Time fraction: The Start Time Fraction field shall, in units of 1/256 s, indicate an offset, delay, from the Start Time field before Stream transfer is to commence.

Stream Consumer: A 24 bit address identifier, indicating the destination of the Available Bit Rate Stream.

Start Time: The Start Time field indicates in System Stream Time seconds, the instant at which Stream transfer is to commence.

Replies to Stream requests:

Stream request rejection

Signifies the rejection of the PRABRS request

– Reject payload; the format of the Reject response is shown in 5.3.6.3.2.

Stream request acceptance

Signifies acceptance of the PRABRS request

– Accept payload; the format of the Accept response is shown in 5.3.6.3.1.3.

5.3.6.1.3 Preemptive Reserve Real–Time Stream

The Preemptive Reserve Real–Time Stream (PRRTS) Stream request is used by a Stream Definer to request reservation for the transfer of one or more Sequences, in a periodic Real–Time Time Slot pattern, from a Stream Producer, to a given destination, the Stream Consumer, starting at a specific time.

The System Resource Manager will, to successfully honor the Stream request, terminate lower priority Available Bit Rate or Real–Time Streams, blocking the PRRTS.

The Stream Definer, need not be the Stream Producer nor the Stream Consumer.

The Stream Consumer, may for Class 3 Real–Time Streams be a Multicast group or the well-known Broadcast address hex ‘FFFFFF’.

Payload: The format of the PRRTS Stream request is shown in table 22.

Table 22 – PRRTS (07)

Item	Size (Bytes)
Control	1
Stream Command Code (hex ‘07’)	1
Quality of Service Parameters	1
Transaction Identifier	1
Producer Tag	1
Stream Producer	3
Start Time fraction (1/256 s)	1
Stream Consumer	3
Start Time (s)	4

Control: An unsigned 8 bit field, consisting of two 4 bit sub-fields, a Priority field and a Class of Service field. The 4 bit Priority field is limited to the range of 0, highest priority, to 7, the lowest priority level. The Priority field uses bits 28 through 31 of the first word in the SCS request. The second field, using bits 24 through 27 in the first word, indicate the Class of Service which will be used to transmit the requested Stream. The encoding of the Class of Service field is as shown in table 15.

Stream Command Code: An 8 bit field uniquely identifying the Stream request. The SCC for the PRRTS Stream request shall be hex ‘07’.

Quality of Service Parameters: The 8 bit Quality of Service Parameter field define either a specific predefined set of Quality of Service Parameter (see 5.2) or form a pair of sub-fields specifying Jitter and Bandwidth requirements. The most significant bit, bit 15, in the Quality of Service Parameter field is used to determine, whether predefined Quality of Service parameters or separate Jitter and Bandwidth is being specified. If bit 15 is set = 1, then predefined Quality of Service, as defined in 5.2, is being requested. If bit 15 is set = 0 then Jitter and Bandwidth requirements are provided.

The unsigned 2 bit Jitter field, using bits 13 through 14, in the first word of the Stream request, define the maximum amount of source jitter, the System Resource Manager may introduce by means of Time Slot scheduling. Table 23 defines the maximum amount of Time Slot scheduling jitter the System Resource Manager may introduce for different values of the Jitter field.

Table 23 – SRM Jitter limit – Encoding

Jitter limit	Value
Source Jitter ≤ 1 Time Slot	0
Source Jitter ≤ 2 Time Slot	1
Source Jitter ≤ 4 Time Slot	2
Source Jitter ≤ 6 Time Slot	3

The unsigned 5 bit Bandwidth field (N), using bits 8 through 12, in the first word of the Stream request, define the bandwidth requirement for the Real–Time Stream, in units of Time Slot per second.

$$N = \begin{cases} \text{Ceiling}\left(\dfrac{\text{Bandwidth (B/s)}}{1\,350\,000}\right) & \text{@ 1,062 5 GBaud} \\ \text{Ceiling}\left(\dfrac{\text{Bandwidth (B/s)}}{1\,151\,000}\right) & \text{@ 2,125 GBaud} \\ \text{Ceiling}\left(\dfrac{\text{Bandwidth (B/s)}}{753\,000}\right) & \text{@ 4,25 GBaud} \end{cases}$$

$$31 \geq N \geq 1$$

Transaction Identifier: An 8 bit identifier, assigned by the Stream Definer. The Transaction Identifier ensure that the Stream Definer can match the Accept or Reject Stream reply to the initial PRRTS Stream request.

Producer Tag: An unique 8 bit identifier, assigned by the Stream Producer to a specific Audio Visual piece of material or source, selected by a method outside the scope of this document.

Stream Producer: A 24 bit address identifier, indicating the source of the Real–Time Stream.

Start Time fraction: The Start Time Fraction field shall, in units of 1/256 s, indicate an offset, delay, from the Start Time field before Stream transfer is to commence.

Stream Consumer: A 24 bit address identifier, indicating the destination of the Real–Time Stream.

Start Time: The Start Time field indicates in System Stream Time seconds, the instant at which Stream transfer is to commence.

Replies to Stream requests:

Stream request rejection

Signifies the rejection of the PRRTS request

– Reject payload; the format of the Reject response is shown in 5.3.6.3.2.

Stream request acceptance

Signifies acceptance of the PRRTS request

– Accept payload; the format of the Accept response is shown in 5.3.6.3.1.4.

5.3.6.1.4 Reserve Asynchronous Stream

The Reserve Asynchronous Stream (RAS) Stream request is used by a Stream Producer, the Stream Definer, to request reservation for the transfer of one or more Sequences, to a given destination, the Stream Consumer, in a single Time Slot.

Payload: The format of the RAS Stream request is shown in table 24.

Table 24 – RAS (01)

Item	Size (Bytes)
Control	1
Stream Command Code (hex ‘01’)	1
Reserved	1
Transaction Identifier	1
Reserved	1
Reserved	3
Reserved	1
Stream Consumer	3
Number of Sequences	2
Total Number of Data Frames	2

Control: An unsigned 8 bit field, consisting of two 4 bit sub-fields, a Priority field and a Class of Service field. The 4 bit Priority field is limited to the range of 0, highest priority, to 7, the lowest priority level. The Priority field uses bits 28 through 31 of the first word in the SCS request. The second field, using bits 24 through 27 in the first word, indicate the Class of Service which will be used to transmit the requested Stream. The encoding of the Class of Service field is as shown in table 15.

Stream Command Code: An 8 bit field uniquely identifying the Stream request. The SCC for the RAS Stream request shall be hex ‘01’.

Transaction Identifier: An 8 bit identifier, assigned by the Stream Definer. The Transaction Identifier ensure that the Stream Definer can match the Accept or Reject Stream reply to the initial RAS Stream request.

Stream Consumer: A 24 bit address identifier, indicating the destination of the Asynchronous Stream.

Number of Sequences: The number of Sequences to be transferred from the Stream Producer to the Stream Consumer, in the requested Time Slot.

Total Number of Data Frames: The total number of Data frames for to be transferred from the Stream Producer to the Stream Consumer, in the requested Time Slot.

Replies to Stream requests:

Stream request rejection
Signifies the rejection of the RAS request

– Reject payload; the format of the Reject response is shown in 5.3.6.3.2.

Stream request acceptance
Signifies acceptance of the RAS request

– Accept payload; the format of the Accept response is shown in 5.3.6.3.1.6.

5.3.6.1.5 Reserve Available Bit Rate Stream

The Reserve Available Bit Rate Stream (RABRS) Stream request is used by a Stream Definer to request reservation for the transfer of one or more Sequences, in a variable Available Bit Rate Time Slot pattern, from a Stream Producer, to a given destination, the Stream Consumer, starting at a specific time.

The Stream Definer, need not be the Stream Producer nor the Stream Consumer.

The Stream Consumer, may for Class 3 Available Bit Rate Streams be a Multicast group or the well-known Broadcast address hex ‘FFFFFF’.

Payload: The format of the RABRS Stream request is shown in table 25.

Control: An unsigned 8 bit field, consisting of two 4 bit sub-fields, a Priority field and a Class of Service field. The 4 bit Priority field is limited to the range of 0, highest priority, to 7, the lowest priority level. The Priority field uses bits 28 through 31 of the first word in the SCS request. The second field, using bits 24 through 27 in the first word, indicate the Class of Service which will be used to transmit the requested Stream. The encoding of the Class of Service field is as shown in table 15.

Stream Command Code: An 8 bit field uniquely identifying the Stream request. The SCC for the RABRS Stream request shall be hex ‘02’.

Table 25 – RABRS (02)

Item	Size (Bytes)
Control	1
Stream Command Code (hex ‘02’)	1
Reserved	1
Transaction Identifier	1
Producer Tag	1
Stream Producer	3
Start Time fraction (1/256 s)	1
Stream Consumer	3
Start Time (s)	4

Transaction Identifier: An 8 bit identifier, assigned by the Stream Definer. The Transaction Identifier ensure that the Stream Definer can match the Accept or Reject Stream reply to the initial RABRS Stream request.

Producer Tag: An unique 8 bit identifier, assigned by the Stream Producer to a specific Audio Visual piece of material or source, selected by a method outside the scope of this document.

Stream Producer: A 24 bit address identifier, indicating the source of the Available Bit Rate Stream.

Start Time fraction: The Start Time Fraction field shall, in units of 1/256 s, indicate an offset, delay, from the Start Time field before Stream transfer is to commence.

Stream Consumer: A 24 bit address identifier, indicating the destination of the Available Bit Rate Stream.

Start Time: The Start Time field indicates in System Stream Time seconds, the instant at which Stream transfer is to commence.

Replies to Stream requests:

Stream request rejection

Signifies the rejection of the RABRS request

– Reject payload; the format of the Reject response is shown in 5.3.6.3.2.

Stream request acceptance

Signifies acceptance of the RABRS request

– Accept payload; the format of the Accept response is shown in 5.3.6.3.1.7.

5.3.6.1.6 Reserve Real–Time Stream

The Reserve Real–Time Stream (RRTS) Stream request is used by a Stream Definer to request reservation for the transfer of one or more Sequences, in a periodic Real–Time Time Slot pattern, from a Stream Producer, to a given destination, the Stream Consumer, starting at a specific time.

The Stream Definer, need not be the Stream Producer nor the Stream Consumer.

The Stream Consumer, may for Class 3 Real–Time Streams be a Multicast group or the well-known Broadcast address hex ‘FFFFFF’.

Payload: The format of the RRTS Stream request is shown in table 26.

Table 26 – RRTS (03)

Item	Size (Bytes)
Control	1
Stream Command Code (hex ‘03’)	1
Quality of Service Parameters	1
Transaction Identifier	1
Producer Tag	1
Stream Producer	3
Start Time fraction (1/256 s)	1
Stream Consumer	3
Start Time (s)	4

Control: An unsigned 8 bit field, consisting of two 4 bit sub-fields, a Priority field and a Class of Service field. The 4 bit Priority field is limited to the range of 0, highest priority, to 7, the lowest priority level. The Priority field uses bits 28 through 31 of the first word in the SCS request.

The second field, using bits 24 through 27 in the first word, indicate the Class of Service which will be used to transmit the requested Stream. The encoding of the Class of Service field is as shown in table 15.

Stream Command Code: An 8 bit field uniquely identifying the Stream request. The SCC for the RRTS Stream request shall be hex ‘03’.

Quality of Service Parameters: The 8 bit Quality of Service Parameter field define either a specific predefined set of Quality of Service Parameter (see 5.2) or form a pair of sub-fields specifying Jitter and Bandwidth requirements. The most significant bit, bit 15, in the Quality of Service Parameter field is used to determine, whether predefined Quality of Service parameters or separate Jitter and Bandwidth is being specified. If bit 15 is set = 1, then predefined Quality of Service, as defined in 5.2, is being requested. If bit 15 is set = 0 then Jitter and Bandwidth requirements are provided.

The unsigned 2 bit Jitter field, using bits 13 through 14, in the first word of the Stream request, define the maximum amount of source jitter, the System Resource Manager may introduce by means of Time Slot scheduling. Table 23 defines the maximum amount of Time Slot scheduling jitter the System Resource Manager may introduce for different values of the Jitter field.

$$N = \begin{cases} \text{Ceiling}\left(\dfrac{\text{Bandwidth (B/s)}}{1\ 350\ 000}\right) & \text{@ 1,062 5 GBaud} \\ \text{Ceiling}\left(\dfrac{\text{Bandwidth (B/s)}}{1\ 151\ 000}\right) & \text{@ 2,125 GBaud} \\ \text{Ceiling}\left(\dfrac{\text{Bandwidth (B/s)}}{753\ 000}\right) & \text{@ 4,25 GBaud} \end{cases}$$

$$31 \geq N \geq 1$$

The unsigned 5 bit Bandwidth field (N), using bits 8 through 12, in the first word of the Stream request, define the bandwidth requirement for the Real–Time Stream, in units of Time Slot per second.

Transaction Identifier: An 8 bit identifier, assigned by the Stream Definer. The Transaction Identifier ensure that the Stream Definer can match the Accept or Reject Stream reply to the initial RRTS Stream request.

Producer Tag: An unique 8 bit identifier, assigned by the Stream Producer to a specific Audio Visual piece of material or source, selected by a method outside the scope of this document.

Stream Producer: A 24 bit address identifier, indicating the source of the Real–Time Stream.

Start Time fraction: The Start Time Fraction field shall, in units of 1/256 s, indicate an offset, delay, from the Start Time field before Stream transfer is to commence.

Stream Consumer: A 24 bit address identifier, indicating the destination of the Real–Time Stream.

Start Time: The Start Time field indicates in System Stream Time seconds, the instant at which Stream transfer is to commence.

Replies to Stream requests:

Stream request rejection
Signifies the rejection of the RRTS request

– Reject payload; the format of the Reject response is shown in 5.3.6.3.2.

Stream request acceptance
Signifies acceptance of the RRTS request

– Accept payload; the format of the Accept response is shown in 5.3.6.3.1.8.

5.3.6.1.7 Resume Real–Time Stream

The Resume Real–Time Stream (RSRTS) Stream request is used by a Stream Definer to request resumption of a Real–Time Stream, from a given Stream Producer, at a specific time.

The Stream Definer, need not be the Stream Producer nor the Stream Consumer.

Payload: The format of the RSRTS Stream request is shown in table 27.

Stream Identifier: The unsigned 8 bit identifier, assigned by the System Resource Manager to the Real–Time Stream.

Stream Command Code: An 8 bit field uniquely identifying the Stream request. The SCC for the RSRTS Stream request shall be hex ‘13’.

Transaction Identifier: An 8 bit identifier, assigned by the Stream Definer. The Transaction Identifier ensure that the Stream Definer can

Table 27 – RSRTS (13)

Item	Size (Bytes)
Stream Identifier	1
Stream Command Code (hex ‘13’)	1
Reserved	1
Transaction Identifier	1
Reserved	1
Stream Producer	3
Resume Time fraction (1/256 s)	1
Reserved	3
Resume Time (s)	4

match the Accept or Reject Stream reply to the initial RSRTS Stream request.

Stream Producer: A 24 bit address identifier, indicating the source of the Real–Time Stream.

Resume Time fraction: The Resume Time Fraction field shall, in units of 1/256 s, indicate an offset, delay, from the Resume Time field before Stream transfer is to continue.

Resume Time: The Resume Time field indicates in System Stream Time seconds, the instant at which Stream transfer is to continue.

Replies to Stream requests:

Stream request rejection

Signifies the rejection of the RSRTS request

– Reject payload; the format of the Reject response is shown in 5.3.6.3.2.

Stream request acceptance

Signifies acceptance of the RSRTS request

– Accept payload; the format of the Accept response is shown in 5.3.6.3.1.9.

5.3.6.1.8 Suspend Real–Time Stream

The Suspend Real–Time Stream (SRTS) Stream request is used by a Stream Definer to request quiescence of a Real–Time Stream, from a given Stream Producer, at a specific time.

The Stream Definer, need not be the Stream Producer nor the Stream Consumer.

Payload: The format of the SRTS Stream request is shown in table 28.

Table 28 – SRTS (0B)

Item	Size (Bytes)
Stream Identifier	1
Stream Command Code (hex ‘0B’)	1
Reserved	1
Transaction Identifier	1
Reserved	1
Stream Producer	3
Suspend Time fraction (1/256 s)	1
Reserved	3
Suspend time (s)	4

Stream Identifier: The unsigned 8 bit identifier, assigned by the System Resource Manager to the Real–Time Stream.

Stream Command Code: An 8 bit field uniquely identifying the Stream request. The SCC for the SRTS Stream request shall be hex ‘0B’.

Transaction Identifier: An 8 bit identifier, assigned by the Stream Definer. The Transaction Identifier ensure that the Stream Definer can match the Accept or Reject Stream reply to the initial SRTS Stream request.

Stream Producer: A 24 bit address identifier, indicating the source of the Real–Time Stream.

Suspend Time fraction: The Suspension Time Fraction field shall, in units of 1/256 s, indicate an offset, delay, from the Suspension Time field before Stream transfer is to be suspended.

Suspend Time: The Suspension Time field indicates in System Stream Time seconds, the instant at which Stream transfer is to be suspended.

Replies to Stream requests:

Stream request rejection
Signifies the rejection of the SRTS request

– Reject payload; the format of the Reject response is shown in 5.3.6.3.2.

Stream request acceptance
Signifies acceptance of the SRTS request

– Accept payload; the format of the Accept response is shown in 5.3.6.3.1.11.

5.3.6.1.9 Terminate Available Bit Rate Stream

The Terminate Available Bit Rate Stream (TABRS) Stream request is used by a Stream Definer to request cessation of an Available Bit Rate Stream, from a given Stream Producer, at a specific time.

The Stream Definer, need not be the Stream Producer nor the Stream Consumer.

Payload: The format of the TABRS Stream request is shown in table 29.

Table 29 – TABRS (0E)

Item	Size (Bytes)
Stream Identifier	1
Stream Command Code (hex ‘0E’)	1
Reserved	1
Transaction Identifier	1
Reserved	1
Stream Producer	3
Termination Time fraction (1/256 s)	1
Reserved	3
Termination Time (s)	4

Stream Identifier: The unsigned 8 bit identifier, assigned by the System Resource Manager to the Available Bit Rate Stream.

Stream Command Code: An 8 bit field uniquely identifying the Stream request. The SCC for the TABRS Stream request shall be hex ‘0E’.

Transaction Identifier: An 8 bit identifier, assigned by the Stream Definer. The Transaction Identifier ensure that the Stream Definer can match the Accept or Reject Stream reply to the initial TABRS Stream request.

Stream Producer: A 24 bit address identifier, indicating the source of the Available Bit Rate Stream.

Termination Time fraction: The Termination Time Fraction field shall, in units of 1/256 s, indicate an offset, delay, from the Termination Time field before Stream transfer is to cease.

Termination Time: The Termination Time field indicates in System Stream Time seconds, the instant at which Stream transfer is to cease.

Replies to Stream requests:

Stream request rejection
Signifies the rejection of the TABRS request

– Reject payload; the format of the Reject response is shown in 5.3.6.3.2.

Stream request acceptance
Signifies acceptance of the TABRS request

– Accept payload; the format of the Accept response is shown in 5.3.6.3.1.13.

5.3.6.1.10 Terminate Real–Time Stream

The Terminate Real–Time Stream (TRTS) Stream request is used by a Stream Definer to request cessation of a Real–Time Stream, from a given Stream Producer, at a specific time.

The Stream Definer, need not be the Stream Producer nor the Stream Consumer.

Payload: The format of the TRTS Stream request is shown in table 30.

Stream Identifier: The unsigned 8 bit identifier, assigned by the System Resource Manager to the Real–Time Stream.

Stream Command Code: An 8 bit field uniquely identifying the Stream request. The SCC for the TRTS Stream request shall be hex ‘0F’.

Transaction Identifier: An 8 bit identifier, assigned by the Stream Definer. The Transaction Identifier ensure that the Stream Definer can match the Accept or Reject Stream reply to the initial TRTS Stream request.

Table 30 – TRTS (0F)

Item	Size (Bytes)
Stream Identifier	1
Stream Command Code (hex ‘0F’)	1
Reserved	1
Transaction Identifier	1
Reserved	1
Stream Producer	3
Termination Time fraction (1/256 s)	1
Reserved	3
Termination Time (s)	4

Stream Producer: A 24 bit address identifier, indicating the source of the Real–Time Stream.

Termination Time fraction: The Termination Time Fraction field shall, in units of 1/256 s, indicate an offset, delay, from the Termination Time field before Stream transfer is to cease.

Termination Time: The Termination Time field indicates in System Stream Time seconds, the instant at which Stream transfer is to cease.

Replies to Stream requests:

Stream request rejection
Signifies the rejection of the TRTS request

– Reject payload; the format of the Reject response is shown in 5.3.6.3.2.

Stream request acceptance
Signifies acceptance of the TRTS request

– Accept payload; the format of the Accept response is shown in 5.3.6.3.1.15.

5.3.6.2 Stream Control Service – Notifications

SCS notifications are used by the System Resource Manager to inform the Stream Producer of Stream request made by a Stream Definer, other than the Stream Producer.

The following SCS notification requests are defined:

– *Available Bit Rate Stream Notification (ABRSN):* Used by the System Resource Manager to prepare the Stream Producer to send an Available Bit Rate Stream.

– *Real–Time Stream Notification (RTSN):* Used by the System Resource Manager to prepare the Stream Producer to send a Real–Time Stream.

– *Resume Real–Time Stream Notification (RSRTSN):* Used by the System Resource Manager to prepare the Stream Producer to resume transmission of a Real–Time Stream.

– *Suspend Real–Time Stream Notification (SRTSN):* Used by the System Resource Manager to inform the Stream Producer to suspend transmission of a Real–Time Stream.

– *Terminate Available Bit Rate Stream Notification (TABRSN):* Used by the System Resource Manager to inform the Stream Producer to stop transmission of an Available Bit Rate Stream.

– *Terminate Real–Time Stream Notification (TRTSN):* Used by the System Resource Manager to inform the Stream Producer to stop transmission of a Real–Time Stream.

5.3.6.2.1 Available Bit Rate Stream Notification

The Available Bit Rate Stream Notification (ABRSN), is sent by the System Resource Manager to the Stream Producer, unless it was the Stream Definer.

The ABRSN Stream request shall serve to inform the Stream Producer of the creation of an Available Bit Rate Stream at a specific future date.

The ABRSN shall be sent at least 1/16 second before the Stream Producer is required to send Available Bit Rate information for the first Time Slot.

Payload: The format of the ABRSN Stream notification is shown in table 31.

Stream Identifier: The unsigned 8 bit identifier, assigned by the System Resource Manager to the Available Bit Rate Stream. Used by the Stream Producer to locate the Stream Slot Counter.

Table 31 – ABRSN (22)

Item	Size (Bytes)
Stream Identifier	1
Stream Command Code (hex ‘22’)	1
Reserved	1
Transaction Identifier	1
Producer Tag	1
Stream Producer	3
Reserved	1
Stream Consumer	3
Start Time Slot	4

Stream Command Code: An 8 bit field uniquely identifying the Stream request. The SCC for the ABRSN request shall be hex ‘22’.

Transaction Identifier: An 8 bit identifier, assigned by the System Resource Manager. The Transaction Identifier ensure that the System Resource Manager can match the Accept or Reject Stream reply to the ABRSN Stream request.

Producer Tag: An unique 8 bit identifier, assigned by the Stream Producer to a specific Audio Visual piece of material or source, selected by a method outside the scope of this document.

Stream Producer: A 24 bit address identifier, indicating the source of the Available Bit Rate Stream.

Stream Consumer: A 24 bit address identifier, indicating the destination of the Available Bit Rate Stream.

Start Time Slot: The Slot Time, for the Slot Time Counter identified by the Stream Identifier, at which Available Bit Rate transmission is to commence.

The Stream Producer shall set the Stop Timer to its default value (see 5.1.5(z)).

Replies to Stream notifications:

Stream notification rejection

Signifies the rejection of the ABRSN request

– Reject payload; the format of the Reject response is shown in 5.3.6.3.2.

Stream notification acceptance

Signifies acceptance of the ABRSN request

– Accept payload; the format of the Accept response is shown in 5.3.6.3.1.1.

5.3.6.2.2 Real–Time Stream Notification

The Real–Time Stream Notification (RTSN), is sent by the System Resource Manager to the Stream Producer, unless it was the Stream Definer.

The RTSN Stream request shall serve to inform the Stream Producer of the creation of a Real–Time Stream at a specific future date.

The RTSN shall be sent at least 1/16 second before the Stream Producer is required to send Real–Time information for the first Time Slot.

Payload: The format of the RTSN Stream notification is shown in table 32.

Table 32 – RTSN (23)

Item	Size (Bytes)
Stream Identifier	1
Stream Command Code (hex ‘23’)	1
Quality of Service Parameters	1
Transaction Identifier	1
Producer Tag	1
Stream Producer	3
Reserved	1
Stream Consumer	3
Start Time Slot	4

Stream Identifier: The unsigned 8 bit identifier, assigned by the System Resource Manager to the Real–Time Stream. Used by the Stream Producer to locate the Stream Slot Counter.

Stream Command Code: An 8 bit field uniquely identifying the Stream request. The SCC for the RTSN request shall be hex ‘23’.

Transaction Identifier: An 8 bit identifier, assigned by the System Resource Manager. The Transaction Identifier ensure that the System Resource Manager can match the Accept or Reject Stream reply to the RTSN Stream request.

Producer Tag: An unique 8 bit identifier, assigned by the Stream Producer to a specific Audio Visual piece of material or source, selected by a method outside the scope of this document.

Stream Producer: A 24 bit address identifier, indicating the source of the Real–Time Stream.

Stream Consumer: A 24 bit address identifier, indicating the destination of the Real–Time Stream.

Start Time Slot: The Slot Time, for the Slot Time Counter identified by the Stream Identifier, at which Real–Time transmission is to commence.

The Stream Producer shall set the Pause, Resume and Stop Timers to their default values (see 5.1.5(ab), 5.1.5(ac) and 5.1.5(ad)).

Replies to Stream notifications:

Stream notification rejection

Signifies the rejection of the RTSN request

– Reject payload; the format of the Reject response is shown in 5.3.6.3.2.

Stream notification acceptance

Signifies acceptance of the RTSN request

– Accept payload; the format of the Accept response is shown in 5.3.6.3.1.5.

5.3.6.2.3 Resume Real–Time Stream Notification

The Resume Real–Time Stream Notification (RSRTSN), is sent by the System Resource Manager to the Stream Producer, unless it was the Stream Definer.

The RSTRSN Stream request shall serve to inform the Stream Producer of the continuation of a Real–Time Stream at a specific future date.

The RSRTSN shall be sent at least 1/16 second before the Stream Producer is required to send Real–Time information for the first Time Slot.

Payload: The format of the RSRTSN Stream notification is shown in table 33.

Table 33 – RSRTSN (33)

Item	Size (Bytes)
Stream Identifier	1
Stream Command Code (hex ‘33’)	1
Reserved	1
Transaction Identifier	1
Reserved	1
Stream Producer	3
Reserved	4
Resume Time Slot	4

Stream Identifier: The unsigned 8 bit identifier, assigned by the System Resource Manager to the Real–Time.

Stream Command Code: An 8 bit field uniquely identifying the Stream request. The SCC for the RSRTSN request shall be hex ‘33’.

Transaction Identifier: An 8 bit identifier, assigned by the System Resource Manager. The Transaction Identifier ensure that the System Resource Manager can match the Accept or Reject Stream reply to the RSRTSN Stream request.

Stream Producer: A 24 bit address identifier, indicating the source of the Real–Time Stream.

Stream Consumer: A 24 bit address identifier, indicating the destination of the Real–Time Stream.

Resume Time Slot: The Slot Time, for the Slot Time Counter identified by the Stream Identifier, at which Real–Time transmission is to continue.

Replies to Stream notifications:

Stream notification rejection

Signifies the rejection of the RSRTSN request

– Reject payload; the format of the Reject response is shown in 5.3.6.3.2.

Stream notification acceptance

Signifies acceptance of the RSRTSN request

– Accept payload; the format of the Accept response is shown in 5.3.6.3.1.10.

5.3.6.2.4 Suspend Real–Time Stream Notification

The Suspend Real–Time Stream Notification (SRTSN), is sent by the System Resource Manager to the Stream Producer, unless it was the Stream Definer.

The SRTSN Stream request shall serve to inform the Stream Producer of the suspension of a Real–Time Stream at a specific future date.

The SRTSN shall be sent at least 1/16 second before the Stream Producer is required to cease sending Real–Time information for the last Time Slot.

Payload: The format of the SRTSN Stream notification is shown in table 34.

Table 34 – SRTSN (2B)

Item	Size (Bytes)
Stream Identifier	1
Stream Command Code (hex ‘2B’)	1
Reserved	1
Transaction Identifier	1
Reserved	1
Stream Producer	3
Reserved	4
Suspend Time Slot	4

Stream Identifier: The unsigned 8 bit identifier, assigned by the System Resource Manager to the Real–Time Stream.

Stream Command Code: An 8 bit field uniquely identifying the Stream request. The SCC for the SRTSN request shall be hex ‘2B’.

Transaction Identifier: An 8 bit identifier, assigned by the System Resource Manager. The Transaction Identifier ensure that the System Resource Manager can match the Accept of Reject Stream reply to the SRTSN Stream request.

Stream Producer: A 24 bit address identifier, indicating the source of the Real–Time Stream.

Stream Consumer: A 24 bit address identifier, indicating the destination of the Real–Time Stream.

Suspend Time Slot: The Slot Time, for the Slot Time Counter identified by the Stream Identifier, at which Real–Time transmission is to cease.

Replies to Stream notifications:

Stream notification rejection

Signifies the rejection of the SRTSN request

– Reject payload; the format of the Reject response is shown in 5.3.6.3.2.

Stream notification acceptance

Signifies acceptance of the SRTSN request

– Accept payload; the format of the Accept response is shown in 5.3.6.3.1.12.

5.3.6.2.5 Terminate Available Bit Rate Stream Notification

The Terminate Available Bit Rate Stream Notification (TABRSN), is sent by the System Resource Manager to the Stream Producer, unless it was the Stream Definer.

The TABRSN Stream request shall serve to inform the Stream Producer of the cessation of an Available Bit Rate Stream at a specific future date, at which time the resources assigned to the Available Bit Rate Stream are released.

The TABRSN shall be sent at least 1/16 second before the Stream Producer is required to send Available Bit Rate information for the last Time Slot.

Payload: The format of the TABRSN Stream notification is shown in table 35.

Table 35 – TABRSN (2E)

Item	Size (Bytes)
Stream Identifier	1
Stream Command Code (hex ‘2E’)	1
Reserved	1
Transaction Identifier	1
Reserved	1
Stream Producer	3
Reserved	4
Termination Time Slot	4

Stream Identifier: The unsigned 8 bit identifier, assigned by the System Resource Manager to the Available Bit Rate Stream.

Stream Command Code: An 8 bit field uniquely identifying the Stream request. The SCC for the TABRSN request shall be hex ‘2E’.

Transaction Identifier: An 8 bit identifier, assigned by the System Resource Manager. The Transaction Identifier ensure that the System Resource Manager can match the Accept or Reject Stream reply to the TABRSN Stream request.

Stream Producer: A 24 bit address identifier, indicating the source of the Available Bit Rate Stream.

Termination Time Slot: The Slot Time, for the Slot Time Counter identified by the Stream Identifier, at which Available Bit Rate transmission is to finish.

Replies to Stream notifications:

Stream notification rejection

Signifies the rejection of the TABRSN request

– Reject payload; the format of the Reject response is shown in 5.3.6.3.2.

Stream notification acceptance

Signifies acceptance of the TABRSN request

– Accept payload; the format of the Accept response is shown in 5.3.6.3.1.14.

5.3.6.2.6 Terminate Real–Time Stream Notification

The Terminate Real–Time Stream Notification (TRTSN), is sent by the System Resource Manager to the Stream Producer, unless it was the Stream Definer.

The TRTSN Stream request shall serve to inform the Stream Producer of the cessation of a Real–Time Stream at a specific future date, at which time the resources assigned to the Real–Time Stream are released.

The TRTSN shall be sent at least 1/16 second before the Stream Producer is required to send Real–Time information for the last Time Slot.

Payload: The format of the TRTSN Stream notification is shown in table 36.

Table 36 – TRTSN (2F)

Item	Size (Bytes)
Stream Identifier	1
Stream Command Code (hex ‘2F’)	1
Reserved	1
Transaction Identifier	1
Reserved	1
Stream Producer	3
Reserved	4
Termination Time Slot	4

Stream Identifier: The unsigned 8 bit identifier, assigned by the System Resource Manager to the Real–Time Stream.

Stream Command Code: An 8 bit field uniquely identifying the Stream request. The SCC for the TRTSN request shall be hex ‘2F’.

Transaction Identifier: An 8 bit identifier, assigned by the System Resource Manager. The Transaction Identifier ensure that the System Resource Manager can match the Accept or

Reject Stream reply to the TRTSN Stream request.

Stream Producer: A 24 bit address identifier, indicating the source of the Real–Time Stream.

Terminate Time Slot: The Slot Time, for the Slot Time Counter identified by the Stream Identifier, at which Real–Time transmission is to finish.

Replies to Stream notifications:

Stream notification rejection

Signifies the rejection of the TRTSN request

– Reject payload; the format of the Reject response is shown in 5.3.6.3.2.

Stream notification acceptance

Signifies acceptance of the TRTSN request

– Accept payload; the format of the Accept response is shown in 5.3.6.3.1.16.

5.3.6.3 Stream Control Service – Replies

Two types of replies are defined; acceptance and rejection. The acceptance replies shall signify that the SCS Stream request or Stream notification for which it is returned is valid and have been accepted. The rejection reply shall signify that the SCS Stream request or Stream notification for which it is returned is invalid and have been rejected.

5.3.6.3.1 Stream Control Service – Acceptance replies

The following SCS Stream acceptance replies are defined:

– *Accept Available Bit Rate Stream Notification (AABRSN):* Used by s Stream Producer to signal its acceptance of an ABRSN Stream notification;

– *Accept No Operation (ANOP):* Used by the System Resource Manager to signal its acceptance of a NOP Stream request;

– *Accept Preemptive Reserve Available Bit Rate Stream (APRABRS):* Used by the System Resource Manager to signal its acceptance of a PRABRS Stream request;

– *Accept Preemptive Reserve Real–Time Stream (APRRTS):* Used by the System Resource Manager to signal its acceptance of a PRRTS Stream request;

– *Accept Real–Time Stream Notification (ARTSN):* Used by a Stream Producer to signal its acceptance of a RTSN Stream notification;

– *Accept Reserve Asynchronous Stream (ARAS):* Used by the System Resource Manager to signal its acceptance of a RAS Stream request;

– *Accept Reserve Available Bit Rate Stream (ARABRS):* Used by the System Resource Manager to signal its acceptance of a RABRS Stream request;

– *Accept Reserve Real–Time Stream (ARRTS):* Used by the System Resource Manager to signal its acceptance of a RRTS Stream request;

– *Accept Resume Real–Time Stream (ARSRTS):* Used by the System Resource Manager to signal its acceptance of a RSRTS Stream request;

– *Accept Resume Real–Time Stream Notification (ARSRTSN):* Used be a Stream Producer to signal its acceptance of a RSRTSN Stream notification;

– *Accept Suspend Real–Time Stream (ASRTS):* Used by the System Resource Manager to signal its acceptance of a SRTS Stream request;

– *Accept Suspend Real–Time Stream Notification (ASRTSN):* Used by a Stream Producer to signal its acceptance of a SRTSN Stream notification;

– *Accept Terminate Available Bit Rate Stream (ATABRS):* Used by the System Resource Manager to signal its acceptance of a TABRS Stream request;

– *Accept Terminate Available Bit Rate Stream Notification (ATABRSN):* Used by a Stream Producer to signal its acceptance of a TABRSN Stream notification;

- *Accept Terminate Real–Time Stream (ATRTS):* Used by the System Resource Manager to signal its acceptance of a TRTS Stream request;

- *Accept Terminate Real–Time Stream Notification (ATRTSN):* Used by a Stream Producer to signal its acceptance of a TRTSN Stream notification.

5.3.6.3.1.1 Accept – Available Bit Rate Stream Notification

The Accept Available Bit Rate Stream Notification (AABRSN) acceptance reply shall signify that the Stream Producer, sending the AABRSN, has accepted the ABRSN Stream notification, for which it is being returned, having allocated the required resource to handle the Available Bit Rate Stream.

Payload: The format of the Accept to the ABRSN request is shown in table 37.

Table 37 – ABRSN Accept (62)

Item	Size (Bytes)
Reserved	1
Stream Command Code (hex ‘62’)	1
Reserved	1
Transaction Identifier	1
Reserved	12

Stream Command Code: An 8 bit field uniquely identifying the Stream acceptance reply. The SCC for the Accept to the ABRSN Stream notification shall be hex ‘62’.

Transaction Identifier: The 8 bit identifier, assigned by the System Resource Manager to the initial ABRSN Stream notification.

5.3.6.3.1.2 Accept – No Operation

The Accept No Operation Stream (ANOP) acceptance reply shall signify that the System Resource Manager has accepted the NOP Stream request, for which it is being returned.

Payload: The format of the Accept to the NOP request is shown in table 38.

Table 38 – NOP Accept (40)

Item	Size (Bytes)
Reserved	1
Stream Command Code (hex ‘40’)	1
Reserved	1
Transaction Identifier	1
Reserved	12

Stream Command Code: An 8 bit field uniquely identifying the Stream acceptance reply. The SCC for the Accept to the NOP Stream request shall be hex ‘40’.

Transaction Identifier: The 8 bit identifier, assigned by the Stream Definer to the initial NOP Stream request.

5.3.6.3.1.3 Accept – Preemptive Reserve Available Bit Rate Stream

The Accept Preemptive Reserve Available Bit Rate Stream (APRABRS) acceptance reply shall signify that the System Resource Manager has accepted the PRABRS Stream request, for which it is being returned, having allocated the required resource to handle an Available Bit Rate Stream.

Payload: The format of the Accept to the PRABRS request is shown in table 39.

Table 39 – PRABRS Accept (46)

Item	Size (Bytes)
Stream Identifier	1
Stream Command Code (hex ‘46’)	1
Reserved	1
Transaction Identifier	1
Reserved	8
Start Time Slot	4

Stream Identifier: The unsigned 8 bit identifier, assigned by the System Resource Manager to the Available Bit Rate Stream. Used by the Stream Producer to locate the Stream Slot Counter.

Stream Command Code: An 8 bit field uniquely identifying the Stream acceptance reply. The SCC for the Accept to the PRABRS Stream request shall be hex ‘46’.

Transaction Identifier: The 8 bit identifier, assigned by the Stream Definer to the initial PRABRS Stream request.

Start Time Slot: The Slot Time, for the Slot Time Counter identified by the Stream Identifier, at which Available Bit Rate transmission is to commence.

If the initial PRABRS Stream request was sent by the Stream Producer, then it shall set the Stop Timer to its default value (see 5.1.5(z)).

5.3.6.3.1.4 Accept – Preemptive Reserve Real–Time Stream

The Accept Preemptive Reserve Real–Time Stream (APRRTS) acceptance reply shall signify that the System Resource Manager has accepted the PRRTS Stream request, for which it is being returned, having allocated the required resource to handle the Real–Time Stream.

Payload: The format of the Accept to the PRRTS request is shown in table 40.

Table 40 – PRRTS Accept (47)

Item	Size (Bytes)
Stream Identifier	1
Stream Command Code (hex ‘47’)	1
Reserved	1
Transaction Identifier	1
Reserved	8
Start Time Slot	4

Stream Identifier: The unsigned 8 bit identifier, assigned by the System Resource Manager to the Real–Time Stream. Used by the Stream Producer to locate the Stream Slot Counter.

Stream Command Code: An 8 bit field uniquely identifying the Stream acceptance reply. The SCC for the Accept to the PRRTS Stream request shall be hex ‘47’.

Transaction Identifier: The 8 bit identifier, assigned by the Stream Definer to the initial PRRTS Stream request.

Start Time Slot: The Slot Time, for the Slot Time Counter identified by the Stream Identifier, at which Real–Time transmission is to commence.

If the initial PRRTS Stream request was sent by the Stream Producer, then it shall set the Pause, Resume and Stop Timers to their default values (see 5.1.5(ab), 5.1.5(ac) and 5.1.5(ad)).

5.3.6.3.1.5 Accept – Real–Time Stream Notification

The Accept Real–Time Stream Notification (ARTSN) acceptance reply shall signify that the Stream Producer, sending the ARTSN, has accepted the RTSN Stream notification, for which it is being returned, having allocated the required resource to handle the Real–Time Stream.

Payload: The format of the Accept to the RTSN request is shown in table 41.

Table 41 – RTSN Accept (63)

Item	Size (Bytes)
Reserved	1
Stream Command Code (hex ‘63’)	1
Reserved	1
Transaction Identifier	1
Reserved	12

Stream Command Code: An 8 bit field uniquely identifying the Stream acceptance reply. The SCC for the Accept to the RTSN Stream notification shall be hex ‘63’.

Transaction Identifier: The 8 bit identifier, assigned by the System Resource Manager to the initial RTSN Stream notification.

5.3.6.3.1.6 Accept – Reserve Asynchronous Stream

The Accept Reserve Asynchronous Stream (ARAS) acceptance reply shall signify that the System Resource Manager has accepted the RAS Stream request, for which it is being returned, having allocated the required resource to handle the Asynchronous Stream.

Payload: The format of the Accept to the RAS request is shown in table 42.

Table 42 – RAS Accept (41)

Item	Size (Bytes)
Stream Identifier (hex ‘00’)	1
Stream Command Code (hex ‘41’)	1
Reserved	1
Transaction Identifier	1
Reserved	8
Slot Number	4

Stream Identifier: The unsigned 8 bit identifier, assigned by the System Resource Manager to the Asynchronous Stream. Used by the Stream Producer to locate the Stream Slot Counter. The Stream Identifier shall be set = hex ‘00’ for Asynchronous Streams.

Stream Command Code: An 8 bit field uniquely identifying the Stream acceptance reply. The SCC for the Accept to the RAS request shall be hex ‘41’.

Transaction Identifier: The 8 bit identifier, assigned by the Stream Definer to the initial RAS Stream request.

Slot Number: The Slot Number, using Time Slot Counter number 0, assigned by the System Resource Manager for the transfer of the Asynchronous Stream.

5.3.6.3.1.7 Accept – Reserve Available Bit Rate Stream

The Accept Reserve Available Bit Rate Stream (ARABRS) acceptance reply shall signify that the System Resource Manager has accepted the RABRS Stream request, for which it is being returned, having allocated the required resource to handle an Available Bit Rate Stream.

Payload: The format of the Accept to the RABRS request is shown in table 43.

Table 43 – RABRS Accept (42)

Item	Size (Bytes)
Stream Identifier	1
Stream Command Code (hex ‘42’)	1
Reserved	1
Transaction Identifier	1
Reserved	8
Start Time Slot	4

Stream Identifier: The unsigned 8 bit identifier, assigned by the System Resource Manager to the Available Bit Rate Stream. Used by the Stream Producer to locate the Stream Slot Counter.

Stream Command Code: An 8 bit field uniquely identifying the Stream acceptance reply. The SCC for the Accept to the RABRS Stream request shall be hex ‘42’.

Transaction Identifier: The 8 bit identifier, assigned by the Stream Definer to the initial RABRS Stream request.

Start Time Slot: The Slot Time, for the Slot Time Counter identified by the Stream Identifier, at which Available Bit Rate transmission is to commence.

If the initial RABRS Stream request was sent by the Stream Producer, then it shall set the Stop Timer to its default value (see 5.1.5(z)).

5.3.6.3.1.8 Accept – Reserve Real–Time Stream

The Accept Reserve Real–Time Stream (ARRTS) acceptance reply shall signify that the System Resource Manager has accepted the RRTS Stream request, for which it is being returned, having allocated the required resource to handle the Real–Time Stream.

Payload: The format of the Accept to the RRTS request is shown in table 44.

Table 44 – RRTS Accept (43)

Item	Size (Bytes)
Stream Identifier	1
Stream Command Code (hex ‘43’)	1
Reserved	1
Transaction Identifier	1
Reserved	8
Start Time Slot	4

Stream Identifier: The unsigned 8 bit identifier, assigned by the System Resource Manager to the Real–Time Stream. Used by the Stream Producer to locate the Stream Slot Counter.

Stream Command Code: An 8 bit field uniquely identifying the Stream acceptance reply. The SCC for the Accept to the RRTS Stream request shall be hex ‘43’.

Transaction Identifier: The 8 bit identifier, assigned by the Stream Definer to the initial RRTS Stream request.

Start Time Slot: The Slot Time, for the Slot Time Counter identified by the Stream Identifier, at which Real–Time transmission is to commence.

If the initial RRTS Stream request was sent by the Stream Producer, then it shall set the Pause, Resume and Stop Timers to their default values (see 5.1.5(ab), 5.1.5(ac) and 5.1.5(ad)).

5.3.6.3.1.9 Accept – Resume Real–Time Stream

The Accept Resume Real–Time Stream (ARSRTS) acceptance reply shall signify that the System Resource Manager has accepted the RSRTS Stream request, for which it is being returned.

Payload: The format of the Accept to the RSRTS request is shown in table 45.

Stream Identifier: The unsigned 8 bit identifier, assigned by the System Resource Manager to the Real–Time Stream.

Table 45 – RSRTS Accept (53)

Item	Size (Bytes)
Stream Identifier	1
Stream Command Code (hex ‘53’)	1
Reserved	1
Transaction Identifier	1
Reserved	8
Resume Time Slot	4

Stream Command Code: An 8 bit field uniquely identifying the Stream acceptance reply. The SCC for the Accept to the RSRTS Stream request shall be hex ‘53’.

Transaction Identifier: The 8 bit identifier, assigned by the Stream Definer to the initial RSRTS Stream request.

Resume Time Slot: The Slot Time, for the Slot Time Counter identified by the Stream Identifier, at which Real–Time transmission is to continue.

5.3.6.3.1.10 Accept – Resume Real–Time Stream Notification

The Accept Resume Real–Time Stream Notification (ARSRTSN) acceptance reply shall signify that the Stream Producer, sending the ARSRTSN, has accepted the RSRTSN Stream notification, for which it is being returned.

Payload: The format of the Accept to the RSRTSN request is shown in table 46.

Table 46 – RSRTSN Accept (73)

Item	Size (Bytes)
Reserved	1
Stream Command Code (hex ‘73’)	1
Reserved	1
Transaction Identifier	1
Reserved	12

Stream Command Code: An 8 bit field uniquely identifying the Stream acceptance reply. The SCC for the Accept to the RSRTSN Stream notification shall be hex ‘73’.

Transaction Identifier: The 8 bit identifier, assigned by the System Resource Manager to the initial RSRTSN Stream notification.

5.3.6.3.1.11 Accept – Suspend Real–Time Stream

The Accept Suspend Real–Time Stream (ASRTS) acceptance reply shall signify that the System Resource Manager has accepted the SRTS Stream request, for which it is being returned.

Payload: The format of the Accept to the SRTS request is shown in table 47.

Table 47 – SRTS Accept (4B)

Item	Size (Bytes)
Stream Identifier	1
Stream Command Code (hex ‘4B’)	1
Reserved	1
Transaction Identifier	1
Reserved	8
Suspend Time Slot	4

Stream Identifier: The unsigned 8 bit identifier, assigned by the System Resource Manager to the Real–Time Stream.

Stream Command Code: An 8 bit field uniquely identifying the Stream acceptance reply. The SCC for the Accept to the SRTS Stream request shall be hex ‘4B’.

Transaction Identifier: The 8 bit identifier, assigned by the Stream Definer to the initial SRTS Stream request.

Suspend Time Slot: The Slot Time, for the Slot Time Counter identified by the Stream Identifier, at which Real–Time transmission is to cease.

5.3.6.3.1.12 Accept – Suspend Real–Time Stream Notification

The Accept Suspend Real–Time Stream Notification (ASRTSN) acceptance reply shall signify that the Stream Producer, sending the ASRTSN, has accepted the SRTSN Stream notification, for which it is being returned.

Payload: The format of the Accept to the SRTSN request is shown in table 48.

Table 48 – SRTSN Accept (6B)

Item	Size (Bytes)
Reserved	1
Stream Command Code (hex ‘6B’)	1
Reserved	1
Transaction Identifier	1
Reserved	12

Stream Command Code: An 8 bit field uniquely identifying the Stream acceptance reply. The SCC for the Accept to the SRTSN Stream notification shall be hex ‘6B’.

Transaction Identifier: The 8 bit identifier, assigned by the System Resource Manager to the initial SRTSN Stream notification.

5.3.6.3.1.13 Accept – Terminate Available Bit Rate Stream

The Accept Terminate Available Bit Rate Stream (ATABRS) acceptance reply shall signify that the System Resource Manager has accepted the TABRS Stream request, for which it is being returned.

Payload: The format of the Accept to the TABRS request is shown in table 49.

Stream Identifier: The unsigned 8 bit identifier, assigned by the System Resource Manager to the Available Bit Rate Stream.

Stream Command Code: An 8 bit field uniquely identifying the Stream acceptance reply. The SCC for the Accept to the TABRS Stream request shall be hex ‘4E’.

Table 49 – TABRS Accept (4E)

Item	Size (Bytes)
Stream Identifier	1
Stream Command Code (hex ‘4E’)	1
Reserved	1
Transaction Identifier	1
Reserved	8
Termination Time Slot	4

Transaction Identifier: The 8 bit identifier, assigned by the Stream Definer to the initial TABRS Stream request.

Termination Time Slot: The Slot Time, for the Slot Time Counter identified by the Stream Identifier, at which Available Bit Rate transmission is to finish, releasing the resources assigned to the Available Bit Rate Stream.

5.3.6.3.1.14 Accept – Terminate Available Bit Rate Stream Notification

The Accept Terminate Available Bit Rate Stream Notification (ATABRSN) acceptance reply shall signify that the Stream Producer, sending the ATABRSN, has accepted the TABRSN Stream notification, for which it is being returned.

Payload: The format of the Accept to the SRTSN request is shown in table 50.

Table 50 – TABRSN Accept (6E)

Item	Size (Bytes)
Reserved	1
Stream Command Code (hex ‘6E’)	1
Reserved	1
Transaction Identifier	1
Reserved	12

Stream Command Code: An 8 bit field uniquely identifying the Stream acceptance reply. The SCC for the Accept to the TABRSN Stream notification shall be hex ‘6E’.

Transaction Identifier: The 8 bit identifier, assigned by the System Resource Manager to the initial TABRSN Stream notification.

5.3.6.3.1.15 Accept – Terminate Real–Time Stream

The Accept Terminate Real–Time Stream (ATRTS) acceptance reply shall signify that the System Resource Manager has accepted the TRTS Stream request, for which it is being returned.

Payload: The format of the Accept to the TRTS request is shown in table 51.

Table 51 – TRTS Accept (4F)

Item	Size (Bytes)
Stream Identifier	1
Stream Command Code (hex ‘4F’)	1
Reserved	1
Transaction Identifier	1
Reserved	8
Termination Time Slot	4

Stream Identifier: The unsigned 8 bit identifier, assigned by the System Resource Manager to the Real–Time Stream.

Stream Command Code: An 8 bit field uniquely identifying the Stream acceptance reply. The SCC for the Accept to the TRTS Stream request shall be hex ‘4F’.

Transaction Identifier: The 8 bit identifier, assigned by the Stream Definer to the initial TRTS Stream request.

Termination Time Slot: The Slot Time, for the Slot Time Counter identified by the Stream Identifier, at which Real–Time transmission is to finish, releasing the resources assigned to the Real–Time Stream.

5.3.6.3.1.16 Accept – Terminate Real–Time Stream Notification

The Accept Terminate Real–Time Stream Notification (ATRTSN) acceptance reply shall signify that the Stream Producer, sending the ATRTSN, has accepted the TRTSN Stream notification, for which it is being returned.

Payload: The format of the Accept to the TRTSN request is shown in table 52.

Table 52 – TRTSN Accept (6F)

Item	Size (Bytes)
Reserved	1
Stream Command Code (hex ‘6F’)	1
Reserved	1
Transaction Identifier	1
Reserved	12

Stream Command Code: An 8 bit field uniquely identifying the Stream acceptance reply. The SCC for the Accept to the TRTSN Stream notification shall be hex ‘6F’.

Transaction Identifier: The 8 bit identifier, assigned by the System Resource Manager to the initial TRTSN Stream notification.

5.3.6.3.2 Stream Control Service – Rejection reply

The Stream reject (S_RJT) rejection reply shall signify, that the Stream Producer, sending the S_RJT, has rejected the Stream notification, for which it is being returned. The S_RJT rejection reply shall signify, that the System Resource Manager, sending the S_RJT, has rejected the Stream request, for which it is being returned.

Payload: The format of the Reject to a Stream request or Stream notification is shown in table 53.

Reason Code: The unsigned 8 bit Reason Code field identify why a given Stream request or Stream notification could not get honored.

Table 53 – Stream Reject (80)

Item	Size (Bytes)
Reason Code	1
Stream Command Code (hex ‘80’)	1
Reserved	1
Transaction Identifier	1
Reserved	12

The lager valued Reason Code shall be selected if multiple Reason Codes apply.

Table 54 shows the defined Reason Codes.

Stream Command Code: An 8 bit field uniquely identifying the Stream reply. The SCC for the S_RJT rejection reply shall be hex ‘80’.

Transaction Identifier: The 8 bit identifier, assigned by the Stream Definer or the System Resource Manager to the initial Stream request or Stream notification. The Transaction Identifier ensure that the Stream Definer or the System Resource Manager can match the Reject Stream reply to the initial Stream request or notification.

5.3.7 System Resource Manager – Selective Prompt

The System Resource Manager Broadcast, Multicast or Unicast a Selective Prompt Extended Link Service command to indicate an Active Time Slot for a specific Stream.

The System Resource Manager shall Broadcast Selective Prompts issued for Asynchronous Streams[4]. The System Resource Manager should not use Broadcast to transmit Selective Prompts issued for Available Bit Rate or Real–Time Streams.

Multicast Groups may be established such that every Stream Producer, assigned a given Stream Identifier, is a member of a specific Multicast Group. The System Resource Manager should only use Multicast to transmit Selective

---

[4] Broadcasting Selective Prompt Extended Link Service commands for Asynchronous Streams ensure that all Stream Producers and Stream Definers achieve and maintain synchronization with the System Resource Managers System Stream Time.

Table 54 – Reject Reason Codes

Encoded value (Hex)	Description
00	Reserved
01	Invalid Priority
02	Invalid Class of Service
03	Invalid Stream Identifier
04	Invalid Stream Command Code
05	Invalid Jitter
06	Invalid Bandwidth
07	Invalid Real–Time Stream Type
08	Invalid Transaction Identifier
09	Invalid Producer Tag
0A	Invalid Stream Producer
0B	Invalid Start Time fraction
0C	Invalid Termination Time fraction
0D	Invalid Suspend Time fraction
0E	Invalid Termination Time fraction
0F	Invalid Stream Consumer
10	Invalid Number of Sequences
11	Invalid Total Number of Data Frames
12	Invalid Start Time unit
13	Invalid Termination Time unit
14	Invalid Suspend Time unit
15	Invalid Termination Time unit
16	Invalid Start Time Slot
17	Invalid Termination Time Slot
18	Invalid Suspend Time Slot
19	Invalid Termination Time Slot
1A–1F	Reserved
20	Stream Producer unable to source another ABR or Real–Time Stream
21	Stream Producer unable to source requested bandwidth
22	Stream Consumer unable to accept another ABR or Real–Time Stream
23	Stream Consumer unable to accept requested bandwidth
24	Insufficient System bandwidth to honor request
25	Insufficient System Resource Manager resources to queue request
26–FF	Reserved

Prompts, if all Stream Producers are members of the Multicast Group.

Protocol:

None

Selective Prompt request Sequence
(Single frame)

Format: FT_1

Addressing: The S_ID field designates the System Resource Manager. The D_ID field designates the destination of the Selective Prompt Extended Link Service command, a Unicast address identifier, a Multicast Identifier or the well–known Broadcast address hex ‘FFFFFF’.

Payload: The format of the Selective Prompt Extended Link Service command is shown in table 55.

Table 55 – Selective Prompt payload

Item	Size (Bytes)
hex ‘xx 00 00 00’	4
Time Slot Number	4
Stream Time	8
Reserved	3
Stream Identifier	1
Time Slot Count	4

Time Slot Number: An unsigned 32 bit field indicating the current Time Slot Number. Time Slots are numbered consecutively, wrapping from hex ‘FFFFFFFF’ to hex ‘00000000’. The Time Slot Number serves to date SCS requests and notifications.

Stream Time: A 64 bit time value, using the format specified by FC–PH for the Expiration timer, providing the System Stream Time, for the use of Stream Definers and Stream Producers.

Stream Identifier: The unsigned 8 bit identifier, used by a Stream Producer to distinguish a specific Stream.

Time Slot Count: The Time Slot Count, a 32 bit unsigned value, indicate the current Time Slot Value for the Stream.

Reply Extended Link Service Sequence:

6 Class 4 Streams

The following types of Class 4 Streams are defined:

- *Class 4 Asynchronous Stream:* A Stream which, carry information on one or more Exchanges between the Stream Producer and the Stream Consumer, limited to a single concurrent Exchange.

- *Class 4 Available Bit Rate Stream:* A Stream which, carry information, at a rate governed by the Fabric, the Stream Producer and, if ACK_1 is used, the Stream Consumer, on one or more concurrent Exchanges between the Stream Producer and the Stream Consumer.

- *Class 4 Real–Time Stream:* A Stream which, carry Real–Time information, at a rate governed by either the Fabric, the Stream Producer or the Stream Consumer, on one or more concurrent Exchanges between the Stream Producer and the Stream Consumer.

The Class 4 Streams are carried in Class 4 circuits, known as Class 4 Stream circuits. Each Class 4 Stream circuit consist of a pair of counter-directional virtual circuits, each with an individually defined set of Quality of Service parameters.

A Class 4 Stream is in one of the following states:

- *Terminated:* a Class 4 Stream not associated with any Class 4 Stream circuit;

- *Primed:* a Class 4 Stream associated with a Class 4 Stream circuit, but unable to transfer information;

- *Dormant:* an inactive Class 4 Stream;

- *Live:* an active Class 4 Stream.

Four types of Class 4 circuits are defined:

- *Composite Class 4 Stream circuit:* A Class 4 Stream circuit able to carry multiple Asynchronous, Available Bit Rate and Real–Time Streams, in any combination, limited only by the Class 4 Stream Producer and the Class 4 Stream Consumer abilities.

- *Conventional Class 4 circuit:* A Class 4 circuit established directly, through the Quality of Service Facilitator, bypassing the System Resource Manager.

- *Permanent Stream circuit:* A Simple Class 4 Stream circuit used to carry a single Class 4 Asynchronous Stream in both direction between either the System Resource Manager or the Quality of Service Facilitator and another Class 4 Stream Port. Two Permanent Stream circuits are defined:

  1) Permanent Stream # 1, the Permanent Stream circuit between the Quality of Service Facilitator and any other Class 4 Stream Port;

  2) Permanent Stream # 2, the Permanent Stream circuit between the System Resource Manager and any other Class 4 Stream Port, except for the Quality of Service Facilitator.

- *Simple Class 4 Stream circuit:* A Class 4 Stream circuit able to carry a single Class 4 Asynchronous, Available Bit Rate or Real–Time Stream, in either direction.

Support for the Composite Class 4 Stream circuit type is optional. However, support for both the Simple Class 4 Stream circuit and the Permanent Stream circuit type are mandatory.

Use of Class 4 Streams in a Fabric or by an N_Port shall not preclude the use of Class 2 or 3 by either the Fabric or an N_Port, nor the use of Class 4 on a Conventional Class 4 circuit.

6.1 Class 4 Stream behavior

6.1.1 Fabric characteristics

Switches supporting Class 4 Streams shall support the establishment of two Permanent Streams, the first, an individual Permanent Stream between the Quality of Service Facilitator and any Class 4 Stream Port, the second, an individual Permanent Stream between the System Resource Manager and any Class 4 Stream Port.

The Fabric shall for Permanent Streams treat SOFc4 delimited Class 4 frames as SOFi4 delimited frames and EOFdt, EOFdti, EOFrt or EOFrti delimited Class 4 frames as EOFt or EOFni delimited frames.

6.1.1.1 Initialization

Switches shall, for each port on the Switch, register the following with the Name Server, at the well-known address hex ‘FFFFFC’[1)]:

- Port Name;
- Node Name;
- Port Type;
- Class of Service.

6.1.1.2 Active Class 4 Stream support

Bandwidth monitoring and throttling shall be made on the basis of the Payload portion for Class 4 Data Frames only. Control of both a maximum and a minimum bandwidth limit is mandatory.

6.1.2 Quality of Service Facilitator (QoSF)

A Quality of Service Facilitator operates as a Class 4 Stream Producer for information sent on Permanent Streams to either the System Resource Manager or a Class 4 Stream Port, and as a Class 4 Stream Consumer for information received on Permanent Streams from either the System Resource Manager or a Class 4 Stream Port.

The following describe the external Fabric behavior of a Quality of Service Facilitator capable of supporting Class 4 Streams, starting with initialization and concluding with routine Class 4 Stream operations.

6.1.2.1 Initialization

The Quality of Service Facilitator (QoSF) shall establish a pair of counter-directional Permanent Streams, known as Permanent Stream #1, with each Class 4 Stream Port (see 6.2.2, 6.2.1 and 6.2.5):

a) The properties of the Permanent Stream from the Quality of Service Facilitator, the CTR, to a Class 4 Stream Port, the CTI, are:

- *Activation Time:* hex ‘FFFFFFFF’;
- *Activation Time Fraction:* hex ‘FFFFFFFF’, immediate activation;
- *Termination Time:* hex ‘00000000’;
- *Termination Time Fraction:* hex ‘00000000’, undefined termination;
- *Stream Type:* hex ‘1’, Class 4 Asynchronous Stream;
- *Class 4 Stream Producer VC_ID:* hex ‘00’;
- *Class 4 Stream Producer ID:* hex ‘FFFFF9’, the Quality of Service Facilitator;
- *Class 4 Stream Consumer VC_ID:* hex ‘00’;
- *Class 4 Stream Consumer ID:* the 24 bit address of the Class 4 Stream Port with which this Permanent Stream is being established’;
- *Producer Tag:* hex ‘0000’;
- *Stream Tag:* hex ‘0000’;
- *Maximum Class 4 Stream Latency:* 66, approximately 1 ms, for the System Resource Manager and 328, approximately 5 ms, for any other Class 4 Stream Port;
- *Number of Component Streams:* 1;
- *Maximum Class 4 Stream Bandwidth limit:* 100 000 Bytes per second;
- *Minimum Class 4 Stream Bandwidth limit:* 10 000 Bytes per second;
- *Class 4 Stream Pacing controller:* 1, Consumer Pacing;
- *Stream Concurrent Sequences:* 10 for the System Resource Manager and 1 for any other Class 4 Stream Port;
- *Stream End–to–End Credit:* 10 for the System Resource Manager and 1 for any other Class 4 Stream Port;
- *Stream Data Field Size:* 256 Byte;
- *Class 4 Stream state:* Live.

b) The properties of the Permanent Stream from the Class 4 Stream Port, the CTI, to the Quality of Service Facilitator, the CTR, are:

- *Activation Time:* hex ‘FFFFFFFF’;

[1)] A Conventional Class 4 circuit may be employed.

- *Activation Time Fraction:* hex ‘FFFFFFFF’, immediate activation;
- *Termination Time:* hex ‘00000000’;
- *Termination Time Fraction:* hex ‘00000000’, undefined termination;
- *Stream Type:* hex ‘1’, Class 4 Asynchronous Stream;
- *Class 4 Stream Producer VC_ID:* hex ‘00’;
- *Class 4 Stream Producer ID:* the 24 bit address of the Class 4 Stream Port with which this Permanent Stream is being established;
- *Class 4 Stream Consumer VC_ID:* hex ‘00’;
- *Class 4 Stream Consumer ID:* hex ‘FFFFF9’, the Quality of Service Facilitator;
- *Producer Tag:* hex ‘0000’;
- *Stream Tag:* hex ‘0000’;
- *Maximum Class 4 Stream Latency:* 66, approximately 1 ms, for the System Resource Manager and 328, approximately 5 ms, for any other Class 4 Stream Port;
- *Number of Component Streams:* 1;
- *Maximum Class 4 Stream Bandwidth limit:* 100 000 Bytes per second;
- *Minimum Class 4 Stream Bandwidth limit:* 10 000 Bytes per second;
- *Class 4 Stream Pacing controller:* 1, Consumer Pacing;
- *Stream Concurrent Sequences:* 10 for the System Resource Manager and 1 for any other Class 4 Stream Port;
- *Stream End–to–End Credit:* 10 for the System Resource Manager and 1 for any other Class 4 Stream Port;
- *Stream Data Field Size:* 256 Byte;
- *Class 4 Stream state:* Live.

The QoSF shall accept the first valid Register System Resource Manager Extended Link Service command it receives on Permanent Stream #1 (see 6.2.8).

6.1.2.2 Active Class 4 Stream support

The QoSF shall for each valid Class 4 Stream circuit Control Extended Link Service commands, received on Permanent Stream #1 (see 6.2.3):

a) establish the conditions specified in the C4_Circuit command, within 1/32 of a second;

b) return an ACK_1 frame to the System Resource Manager, within 1/256 of a second;

c) return an Accept to the System Resource Manager, within 1/16 of a second.

The QoSF shall send a Class 4 Stream circuit End Extended Link Service command, on Permanent Stream #1, within 1/32 of a second of the release of a Class 4 Stream circuit (see 6.2.4).

The QoSF shall respond to valid Get Class 4 Stream Ability Extended Link Service commands, sent by the System Resource Manager and received on Permanent Stream #1 (see 6.2.6).

The QoSF shall for Permanent Streams treat SOFc4 delimited Class 4 frames as SOFi4 delimited frames and EOFdt, EOFdti, EOFrt or EOFrti delimited Class 4 frames as EOFt or EOFni delimited frames.

6.1.3 Stream Consumer

The following describe the external behavior of a Class 4 Stream Consumer, starting with initialization and concluding with routine Class 4 Stream operations.

6.1.3.1 Initialization

A Stream Consumer shall register the following with the Name Server, at the well-known address hex ‘FFFFFC’[1]:

- FC–4 types;
- Port Type;
- and may reregister, replacing the Fabric initiated registration, the:
  - Port Name;

- Node Name;
- Class of Service.

A Class 4 Stream Consumer shall accept the first valid Get Class 4 Stream Ability Extended Link Service command it receives on Permanent Stream #2[2)].

6.1.3.2 Active Class 4 Stream support

A Class 4 Stream Consumer shall for each valid Class 4 Stream Notify Extended Link Service commands, received on Permanent Stream #2 (see 6.2.5):

a) be ready to receive the Class 4 Stream identified in the C4_Notify command, within 1/32 of a second;

b) return an ACK_1 frame to the System Resource Manager, within 1/256 of a second;

c) return an Accept to the System Resource Manager, within 1/64 of a second.

A Class 4 Stream Consumer shall respond to all valid Get Class 4 Stream Ability Extended Link Service command it receives, on Permanent Stream #2, from the System Resource Manager (see 6.2.6).

A Class 4 Stream Consumer shall for Class 4 Streams, follow the rules defined by FC–PH–2 for Class 4 frame flow, and the limits implied by the Get Class 4 Stream Ability parameters. However, the Class 4 Stream Consumer shall never honor a request to deactivate or remove a Permanent Stream, nor shall it request deactivation or removal of a Permanent Stream.

ACK_1 shall be used for all Class 4 Streams which employ Consumer Pacing, ACK_0 may be used for Class 4 Streams which employ Fabric Pacing or Producer Pacing. ACK_N shall never be used for Class 4 Streams.

A Class 4 Stream Consumer shall for Permanent Streams treat SOFc4 delimited Class 4 frames as SOFi4 delimited frames and EOFdt, EOFdti, EOFrt or EOFrti delimited Class 4 frames as EOFt or EOFni delimited frames.

6.1.4 Stream Definer

A Class 4 Stream Definer operates as a Class 4 Stream Producer for information sent on Permanent Streams to either the System Resource Manager or the Quality of Service Facilitator, and as a Class 4 Stream Consumer for information received on Permanent Streams from either the System Resource Manager or the Quality of Service Facilitator.

The following describe the external behavior of a Class 4 Stream Definer, starting with initialization and concluding with routine Class 4 Stream operations.

6.1.4.1 Initialization

A Stream Definer shall register the following with the Name Server, at the well-known address hex ‘FFFFFC’[1)]:

- FC–4 types;
- Port Type;
- and may reregister, replacing the Fabric initiated registration, the:
  - Port Name;
  - Node Name;
  - Class of Service.

A Class 4 Stream Definer shall accept the first valid Get Class 4 Stream Ability Extended Link Service command it receives on Permanent Stream #2.

6.1.4.2 Active Class 4 Stream support

A Class 4 Stream Definer shall send a Class 4 Get Tag Extended Link Service command prior to sending any C4_Play, C4_Pause or C4_Stop commands for a Class 4 Stream (see 6.2.2).

A Class 4 Stream Definer shall send Begin Stream (C4_Play) Extended Link Service commands at least 1/16 of a second prior to the initial activation of a Class 4 Stream (see 6.2.1).

A Class 4 Stream Definer may send Suspend Stream (C4_Pause) Extended Link Service commands, if it does, then they shall be sent at least 1/16 of a second prior to the suspension of the targeted Class 4 Stream (see 6.2.9).

A Class 4 Stream Definer shall send Terminate Stream (C4_Stop) Extended Link Service commands at least 1/16 of a second prior to the deactivation of a Class 4 Stream (see 6.2.11).

[2)] This establishes the identity of the System Resource Manager.

A Class 4 Stream Definer may send Query Tag Extended Link Service commands to the System Resource Manager (see 6.2.7).

A Class 4 Stream Definer may send Tag Query Extended Link Service commands to the System Resource Manager (see 6.2.10).

A Class 4 Stream Definer shall respond to all valid Get Class 4 Stream Ability Extended Link Service commands it receives, on Permanent Stream #2, from the Register System Resource Manager.

A Class 4 Stream Definer shall for Class 4 Streams, follow the rules defined by FC–PH–2 for Class 4 frame flow, and the limits implied by the Get Class 4 Stream Ability parameters. However, the Class 4 Stream Definer shall never honor a request to deactivate or remove a Permanent Stream, nor shall it request deactivation or removal of a Permanent Stream.

ACK_1 shall be used for all Class 4 Streams which employ Consumer Pacing, ACK_0 may be used for Class 4 Streams which employ Fabric Pacing or Producer Pacing. ACK_N shall never be used for Class 4 Streams.

A Class 4 Stream Definer shall for Permanent Streams treat SOFc4 delimited Class 4 frames as SOFi4 delimited frames and EOFdt, EOFdti, EOFrt or EOFrti delimited Class 4 frames as EOFt or EOFni delimited frames.

6.1.5 Stream Producer

A Class 4 Stream Producer operates as a Class 4 Stream Producer for information sent on Permanent Streams to either the System Resource Manager or the Quality of Service Facilitator, and a Class 4 Stream Consumer for information received on Permanent Streams from either the System Resource Manager or the Quality of Service Facilitator.

The following describe the external behavior of a Class 4 Stream Producer, starting with initialization and concluding with routine Class 4 Stream operations.

6.1.5.1 Initialization

A Stream Producer shall register the following with the Name Server, at the well-known address hex ‘FFFFFC’[1)]:

- FC–4 types;
- Port Type;
- and may reregister, replacing the Fabric initiated registration, the:
  - Port Name;
  - Node Name;
  - Class of Service.

A Class 4 Stream Producer shall accept the first valid Get Class 4 Stream Ability Extended Link Service command it receives on Permanent Stream #2 (see 6.2.6).

6.1.5.2 Active Class 4 Stream support

A Class 4 Stream Producer shall for each valid Class 4 Get Tag (see 6.2.2) Extended Link Service command, received on Permanent Stream #2:

a) return an ACK_1 frame to the System Resource Manager, within 1/256 of a second;

b) return an Accept to the System Resource Manager, within 1/32 of a second.

A Class 4 Stream Producer shall for each valid Begin Stream (see 6.2.1) Extended Link Service command, received on Permanent Stream #2, the Class 4 Stream Producer shall accept a maximum of 8 unserviced C4_Play commands:

c) start transmission within 1/32 of a second of the requested activation time;

d) return an ACK_1 frame to the System Resource Manager, within 1/256 of a second;

e) return an Accept to the System Resource Manager, within 1/32 of a second.

A Class 4 Stream Producer shall for each valid Suspend Stream (see 6.2.9) Extended Link Service command, received on Permanent Stream #2, the Class 4 Stream Producer shall accept a maximum of 8 unserviced C4_Pause commands:

f) cease transmission within 1/32 of a second of the requested suspension time;

g) return an ACK_1 frame to the System Resource Manager, within 1/256 of a second;

h) return an Accept to the System Resource Manager, within 1/32 of a second.

A Class 4 Stream Producer shall for each valid Terminate Stream (see 6.2.11) Extended Link Service command, received on Permanent Stream #2:

i) terminate transmission within 1/32 of a second of the requested termination time;

j) return an ACK_1 frame to the System Resource Manager, within 1/256 of a second;

k) return an Accept to the System Resource Manager, within 1/32 of a second. The Class 4 Stream Producer is free to reuse any of the resources allocated to the Class 4 Stream, once it has sent the Accept.

A Class 4 Stream Producer shall respond to all valid Query Tag Extended Link Service commands, sent by a Class 4 Stream Port (see 6.2.7).

A Class 4 Stream Producer shall respond to all valid Tag Query Extended Link Service commands, sent by a Class 4 Stream Port (see 6.2.10).

A Class 4 Stream Producer shall respond to all valid Get Class 4 Stream Ability Extended Link Service command it receives, on Permanent Stream #2, from the Register System Resource Manager (see 6.2.6).

A Class 4 Stream Producer shall for Class 4 Streams, follow the rules defined by FC–PH–2 for Class 4 frame flow, and the limits implied by the Get Class 4 Stream Ability parameters. However, the Class 4 Stream Producer shall never honor a request to deactivate or remove a Permanent Stream, nor shall it request deactivation or removal of a Permanent Stream.

A Class 4 Stream Producer shall for Permanent Streams treat SOFc4 delimited Class 4 frames as SOFi4 delimited frames and EOFdt, EOFdti, EOFrt or EOFrti delimited Class 4 frames as EOFt or EOFni delimited frames.

A Class 4 Stream Producer shall request removal of a Class 4 Stream circuit, once the last Class 4 Stream, using the Class 4 Stream circuit, is ready for termination. A Class 4 Stream Producer should request deactivation of a Class 4 Stream circuit, once the last Class 4 Stream for the Class 4 Stream circuit is ready to go Dormant.

6.1.6 System Resource Manager

The System Resource Manager operates as a Class 4 Stream Producer for information sent on Permanent Streams to either a Class 4 Stream Port or the Quality of Service Facilitator, and as a Class 4 Stream Consumer for information received on Permanent Streams from either a Class 4 Stream Port or the Quality of Service Facilitator.

The following describe the external behavior of a System Resource Manager, starting with initialization and concluding with routine Class 4 Stream operations.

6.1.6.1 Initialization

A Class 4 Stream Port shall become functional as System Resource Manager if it successfully completes the Register System Resource Manager function (see 6.2.8).

The System Resource Manager shall register the following with the Name Server, at the well-known address hex ‘FFFFFC’1):

– FC–4 types;

– Port Type;

– and may reregister, replacing the Fabric initiated registration, the:

  – Port Name;

  – Node Name;

  – Class of Service.

The System Resource Manager shall collect the Class 4 Stream Resource for the Fabric, using the Get Class 4 Stream Ability (see 6.2.6) Extended Link Service command.

The System Resource Manager shall establish a pair of Permanent Streams, known as Permanent Stream #2, with each Class 4 Stream Port, except for the QoSF, which is served by Permanent Stream #1 (see 6.2.2, 6.2.1, 6.2.5 and 6.1.2):

a) The properties of the Permanent Stream from the System Resource Manager, the CTR, to a Class 4 Stream Port, the CTI, are:

– *Activation Time:* hex ‘FFFFFFFF’;

– *Activation Time Fraction:* hex ‘FFFFFFFF’, immediate activation;

- *Termination Time:* hex ‘00000000’;
- *Termination Time Fraction:* hex ‘00000000’, undefined termination;
- *Stream Type:* hex ‘1’, Class 4 Asynchronous Stream;
- *Class 4 Stream Producer VC_ID:* hex ‘FF’;
- *Class 4 Stream Producer ID:* the address of the System Resource Manager;
- *Class 4 Stream Consumer VC_ID:* hex ‘FF’;
- *Class 4 Stream Consumer ID:* the 24 bit address of the Class 4 Stream Port with which this Permanent Stream is being established’;
- *Producer Tag:* hex ‘0000’;
- *Stream Tag:* hex ‘0000’;
- *Maximum Class 4 Stream Latency:* 328, approximately 5 ms;
- *Number of Component Streams:* 1;
- *Maximum Class 4 Stream Bandwidth limit:* 100 000 Bytes per second;
- *Minimum Class 4 Stream Bandwidth limit:* 10 000 Bytes per second;
- *Class 4 Stream Pacing controller:* 1, Consumer Pacing;
- *Stream Concurrent Sequences:* 5;
- *Stream End–to–End Credit:* 5;
- *Stream Data Field Size:* 256 Byte;
- *Class 4 Stream state:* Live.

b) The properties of the Permanent Stream from the Class 4 Stream Port, the CTI, to the System Resource Manager, the CTR, are:

- *Activation Time:* hex ‘FFFFFFFF’;
- *Activation Time Fraction:* hex ‘FFFFFFFF’, immediate activation;
- *Termination Time:* hex ‘00000000’;
- *Termination Time Fraction:* hex ‘00000000’, undefined termination;
- *Stream Type:* hex ‘1’, Class 4 Asynchronous Stream;
- *Class 4 Stream Producer VC_ID:* hex ‘FF’;
- *Class 4 Stream Producer ID:* the 24 bit address of the Class 4 Stream Port with which this Permanent Stream is being established;
- *Class 4 Stream Consumer VC_ID:* hex ‘FF’;
- *Class 4 Stream Consumer ID:* the address of the System Resource Manager;
- *Producer Tag:* hex ‘0000’;
- *Stream Tag:* hex ‘0000’;
- *Maximum Class 4 Stream Latency:* 328, approximately 5 ms;
- *Number of Component Streams:* 1;
- *Maximum Class 4 Stream Bandwidth limit:* 100 000 Bytes per second;
- *Minimum Class 4 Stream Bandwidth limit:* 10 000 Bytes per second;
- *Class 4 Stream Pacing controller:* 1, Consumer Pacing;
- *Stream Concurrent Sequences:* 5;
- *Stream End–to–End Credit:* 5;
- *Stream Data Field Size:* 256 Byte;
- *Class 4 Stream state:* Live.

The System Resource Manager shall collect the Class 4 Stream Resource for the attached Class 4 Stream Ports, using the Get Class 4 Stream Ability Extended Link Service command (see 6.2.6).

6.1.6.2 Active Class 4 Stream support

The System Resource Manager shall send Class 4 Stream circuit Control Extended Link Service commands at least 1/16 of a second prior to the activation of the Class 4 Stream circuit (see 6.2.3).

The System Resource Manager shall assign Class 4 Streams requesting Fabric Pacing or Producer Pacing to Simple Class 4 Stream circuits.

The System Resource Manager shall send Class 4 Stream Notify Extended Link Service commands at least 1/32 of a second prior to the initial activation of a Class 4 Stream (see 6.2.5).

The System Resource Manager shall for each valid Begin Stream (see 6.2.1) Extended Link Service command, received on Permanent Stream #2:

a) send a C4_Play command to the Class 4 Stream Producer, within 1/16 of a second. The System Resource Manager shall await an Accept to the C4_Notify command it sent to the Class 4 Stream Consumer before it sends the C4_Play command (see 6.2.5)[3)].

b) return an ACK_1 frame to the Class 4 Stream Definer, within 1/256 of a second

c) return an Accept to the Class 4 Stream Definer, within 1/16 of a second. The System Resource Manager shall await an Accept to the C4_Play command it sent to the Class 4 Stream Producer (see 6.1.6.2(a)) before it returns the Accept.

The System Resource Manager shall for each valid Suspend Stream (see 6.2.9) Extended Link Service command, received on Permanent Stream #2:

d) send a C4_Pause command to the Class 4 Stream Producer, within 1/32 of a second;

e) return an ACK_1 frame to the Class 4 Stream Definer, within 1/256 of a second;

f) return an Accept to the Class 4 Stream Definer, within 1/16 of a second. The System Resource Manager shall await an Accept to the C4_Pause command it sent to the Class 4 Stream Producer (see 6.1.6.2(d)) before it returns the Accept.

The System Resource Manager shall for each valid Terminate Stream (see 6.2.11) Extended Link Service command, received on Permanent Stream #2:

g) send a C4_Stop command to the Class 4 Stream Producer, within 1/32 of a second;

h) return an ACK_1 frame to the Class 4 Stream Definer, within 1/256 of a second;

i) return an Accept to the Class 4 Stream Definer, within 1/16 of a second. The System Resource Manager shall await an Accept to the C4_Stop command it sent to the Class 4 Stream Producer (see 6.1.6.2(g)) before it returns the Accept.

The System Resource Manager shall for each valid Class 4 Get Tag Extended Link Service command, received on Permanent Stream #2:

j) send a C4_Tag command to the Class 4 Stream Producer, within 1/32 of a second;

k) return an ACK_1 frame to the Class 4 Stream Definer, within 1/256 of a second;

l) return an Accept to the Class 4 Stream Definer, within 1/16 of a second (see 6.2.2). The System Resource Manager shall await an Accept to the C4_Tag command it sent to the Class 4 Stream Producer (see 6.1.6.2(j)) before it returns the Accept.

The System Resource Manager shall respond to a valid Class 4 Stream circuit End Extended Link Service command, received on Permanent Stream #1, returning an ACK_1 frame within 1/256 of a second. (See 6.2.4)

The System Resource Manager shall respond to a valid Get Class 4 Stream Ability Extended Link Service commands, sent by a Class 4 Stream Definers and received on Permanent Stream #2 (see 6.2.6).

The System Resource Manager shall respond to a valid Query Tag Extended Link Service commands, sent by a Class 4 Stream Port and received on Permanent Stream #2 (see 6.2.7).

The System Resource Manager shall respond to a valid Tag Query Extended Link Service commands, sent by Class 4 Stream Ports and received on Permanent Stream #2 (see 6.2.10).

The System Resource Manager shall for Permanent Streams treat SOFc4 delimited Class 4 frames as SOFi4 delimited frames and EOFdt, EOFdti, EOFrt or EOFrti delimited Class 4 frames as EOFt or EOFni delimited frames.

6.2 Link Services for Class 4 Streams

The following Extended Link Service commands are defined for the support of Class 4 Streams:

[3)] A C4_Notify command is only sent the first time a Class 4 Stream is started.

– *Begin Stream (C4_Play):* Enables a Stream Definer, through the System Resource Manager, to start or restart a Class 4 Stream (see 6.2.1);

– *Class 4 Get Tag (C4_Tag):* Enables a Stream Definer to create a Class 4 Stream, obtaining a Stream Tag (handle) for a unique piece of Audio–Visual material or a specific Audio–Visual source within a Stream Producer (see 6.2.2);

– *Class 4 Stream circuit Control (C4_Circuit):* Lets the System Resource Manager allocate the Fabric resources for a Class 4 Stream circuit (see 6.2.3).

– *Class 4 Stream circuit End (C4_End):* Enables the Quality of Service Facilitator to notify the System Resource Manager of the cessation of a Class 4 Stream circuit (see 6.2.4).

– *Class 4 Stream Notify (C4_Notify):* Enables the System Resource Manager to inform the Class 4 Stream Consumer of a new Class 4 Stream destined for it (see 6.2.5).

– *Get Class 4 Stream Ability (C4_Ability):* Enables a Port to discover the Class 4 Stream abilities of another Port (see 6.2.6);

– *Query Tag:* Enables a Port to discover the Tags assigned to Class 4 Streams for a given Class 4 Stream Producer (see 6.2.7);

– *Register System Resource Manager (RSRM):* Enables a Class 4 Stream Port to become the System Resource Manager (see 6.2.8);

– *Suspend Stream (C4_Pause):* Enables a Stream Definer, through the System Resource Manager, to suspend operation of a Class 4 Stream (see 6.2.9);

– *Tag Query:* Enables a Port to find the characteristics for a Class 4 Stream assigned a specific Tag (see 6.2.10);

– *Terminate Stream (C4_Stop):* Enables a Stream Definer, through the System Resource Manager, to terminate, deactivate, a Class 4 Stream (see 6.2.11).

The required level of support for the Extended Link Service commands for Class 4 Streams by the different Class 4 Stream entities are shown

Table 56 – Class 4 Stream Extended Link Service – Support level

	Class 4 Stream Consumer	**Class 4 Stream Definer**	**Class 4 Stream Producer**	**System Resource Manager**	**Quality of Service Facilitator**
C4_Play	NA	M	M	M	NA
C4_Tag	NA	M	M	M	NA
C4_Circuit	NA	NA	NA	M	M
C4_End	NA	NA	NA	M	M
C4_Notify	M	NA	NA	M	NA
C4_Ability	M	M	M	M	M
Query_Tag	NA	O	M	M	NA
RSRM	NA	NA	NA	M	M
C4_Pause	NA	O	M	M	NA
Tag_Query	NA	O	M	M	NA
C4_Stop	NA	M	M	M	NA
NANot applicable MMandatory O Optional					

6.2.1 Begin Stream

Class 4 Stream Definers and the System Resource Manager sends the Begin Stream (C4_Play) Extended Link Service command to activate a Class 4 Stream, immediately or at a future date.

The C4_Play command shall travel on Permanent Stream #2.

Protocol:

Single frame Begin Stream request Sequence
Single frame Accept reply Sequence

Format: FT_1

Addressing: The S_ID shall designate a Class 4 Stream Definer or the System Resource Manager. The D_ID shall designate the System Resource Manager, if the C4_Play command is sent by a Class 4 Stream Definer. If the C4_Play command is sent by the System Resource Manager, then the D_ID shall designate the Class 4 Stream Producer.

Payload: The format of the C4_Play Extended Link Service command is shown in table 57.

Table 57 – C4_Play Payload

Item	Size (Bytes)
hex ‘xx 00 00 00’	4
Class 4 Stream Producer VC_ID	1
Class 4 Stream Producer ID	3
Class 4 Stream Consumer VC_ID	1
Reserved	1
Tag	2
Activation Time	4
Activation Time Fraction	4

Class 4 Stream Producer VC_ID: This unsigned 8 bit field shall identify the virtual circuit identifier, assigned by the System Resource Manager to the virtual circuit in the direction from the Class 4 Stream Producer to the Class 4 Stream Consumer.

This field shall be set = 0, on any C4_Play command not sent by the System Resource Manager.

Class 4 Stream Producer ID: This unsigned 24 bit field shall identify the source of the Class 4 Stream to be activated.

Class 4 Stream Consumer VC_ID: This unsigned 8 bit field shall designate the virtual circuit identifier, assigned by the System Resource Manager to the virtual circuit in the direction from the Class 4 Stream Consumer to the Class 4 Stream Producer.

This field shall be set = 0, on any C4_Play command not sent by the System Resource Manager.

Tag: An unsigned 16 bit field identifying the Class 4 Stream to be activated.

The Tag field shall be set to the value of the Stream Tag, if the C4_Play command is sent by the Class 4 Stream Definer.

The Tag field shall be set to the value of the Producer Tag, if the C4_Play command is sent by the System Resource Manager.

Activation Time: This unsigned 32 bit field shall identify the time in seconds, relative to 00:00:00 Universal Time (UT) on 1 January 1900, at which transmission for the Class 4 Stream is to begin or resume.

Activation Time Fraction: This unsigned 32 bit field shall identify an offset, in units of 1/4 294 967 297 second, to the Activation Time.

An immediate start or resumption of the Class 4 Stream is requested, if both the ‘Activation Time’ and the ‘Activation Time Fraction’ fields are set = hex ‘FF FF FF FF’.

The Activation Time shall not be set so that both the ‘Activation Time’ and the ‘Activation Time Fraction’ fields are = hex ‘00 00 00 00’.

Reply Extended Link Service Sequence:

Service Reject (LS_RJT)
Signifies the rejection of the C4_Play command

– Reject Reason code

The following Reject Reason codes, in addition to the FC–PH reason codes, are defined:

Invalid Class 4 Stream Producer ID: No Class 4 Stream Port with the given Fibre Channel address exist.

Invalid Tag: The Tag is not recognized by the C4_Play command recipient, i.e. the recipient is not aware of any Class 4 Stream for the identified Class 4 Stream Producer using the given Tag value.

Invalid Activation Time: The time value specified by the ‘Activation Time’ and ‘Activation Time Fraction’ fields is illegal.

Request denied: The C4_Play request have been denied for reason of security or management. However, the Class 4 Stream Producer shall not use this reason to reject a C4_Play command sent by the System Resource Manager.

The Class 4 Stream Producer shall use this reason to reject any C4_Play command not sent by the System Resource Manager.

Accept (ACC)
Signifies acceptance of the C4_Play command

– Accept Payload

The format of the Accept Payload is shown in table 58.

Table 58 – C4_Play Accept Payload

Item	Size (Bytes)
hex ‘02 00 00 00’	4

6.2.2 Class 4 Get Tag

Class 4 Stream Definers sends the Class 4 Get Tag (C4_Tag) Extended Link Service command to the System Resource Manager in order to obtain a unique identifier (Stream Tag) for either a specific piece of Audio–Visual material or Specific Audio–Visual source, for later use by the Class 4 Stream Play, Pause and Stop commands (see 6.2.1, 6.2.9 and 6.2.11).

The System Resource Manager sends the Class 4 Get Tag Extended Link Service command to the Class 4 Stream Producer in order to obtain a unique identifier (Producer Tag) for the material requested by the Class 4 Stream Definer.

The C4_Tag command shall travel on Permanent Stream #2.

Protocol:

Class 4 Get Tag request Sequence
Accept reply Sequence

Format: FT_1

Addressing: The S_ID shall designate a Class 4 Stream Definer or the System Resource Manager. The D_ID shall designate the System Resource Manager, if the C4_Tag command is sent by a Class 4 Stream Definer. If the C4_Tag command is sent by the System Resource Manager, then the D_ID shall designate the Class 4 Stream Producer.

Payload: The format of the C4_Tag Extended Link Service command is shown in table 59.

Table 59 – C4_Tag Payload

Item	Size (Bytes)
hex ‘xx 00 00 00’	4
Activation Time	4
Activation Time Fraction	4
Termination Time	4
Termination Time Fraction	4
Stream Type	1
Class 4 Stream Producer ID	3
Reserved	1
Class 4 Stream Consumer ID	3
Maximum Class 4 Stream Latency	2
Reserved	1
Number of Component Streams	1
Component Stream parameter set #1	k
…	
Component Stream parameter set #n	l

Activation Time: This unsigned 32 bit field shall identify the time in seconds, relative to 00:00:00 Universal Time (UT) on 1 January 1900, at which transmission for the Class 4 Stream is to begin.

Activation Time Fraction: This unsigned 32 bit field shall identify an offset, in units of 1/4 294 967 297 second, to the Activation Time.

No start time for the Class 4 Stream is specified, if both the ‘Activation Time’ and the ‘Activation Time Fraction’ fields are set = hex ‘00 00 00 00’.

The Activation Time shall not be set so that both the ‘Activation Time’ and the ‘Activation Time Fraction’ fields are = hex ‘FF FF FF FF’ (immediate activation).

Termination Time: This unsigned 32 bit field shall identify the time in seconds, relative to 00:00:00 Universal Time (UT) on 1 January 1900, at which transmission for the identified Class 4 Stream is to cease.

Termination Time Fraction: This unsigned 32 bit field shall identify an offset, in units of 1/4 294 967 297 second, to the Termination Time.

No Termination Time for the Class 4 Stream is specified, if both the ‘Termination Time’ and the ‘Termination Time Fraction’ fields are set = hex ‘00 00 00 00’.

The Termination Time shall not be set so that both the ‘Termination Time’ and the ‘Termination Time Fraction’ fields are = hex ‘FF FF FF FF’ (immediate termination).

Stream Type: This unsigned 8 bit field shall identify the Stream Type to be used for this Class 4 Stream.

The following values are defined for the Stream Type field:

- *1:* Class 4 Asynchronous Stream;
- *2:* Class 4 Available Bit Rate Stream;
- *3:* Class 4 Real–Time Stream;
- *other values:* reserved.

Class 4 Stream Producer ID: This unsigned 24 bit field shall indicate the source of the Class 4 Stream for which a Tag is requested.

Class 4 Stream Consumer ID: This unsigned 24 bit field shall indicate the destination of the Class 4 Stream.

Maximum Class 4 Stream Latency: This unsigned 16 bit field shall indicate the longest time, in units of 1/65 536 second, that a Class 4 frame shall spend in the Fabric.

The System Resource Manager may decrease the value initially submitted by the Class 4 Stream Definer.

Number of Component Streams: This unsigned 8 bit field shall identify the number of Component Streams required to carry the Class 4 Stream.

This field shall be set = 1 for Class 4 Asynchronous Streams.

Component Stream parameter set: One or more entries for Component Stream parameters are defined, as specified by the ‘Number of Component Streams’ field. The format of a Component Stream parameter set is shown in table 60.

Table 60 – Component Stream – Parameters

Item	Size (Bytes)
Maximum Class 4 Stream Bandwidth limit	4
Minimum Class 4 Stream Bandwidth limit	4
Stream End–to–End Credit	2
Stream Data Field Size	2
Reserved	2
Length of Component Stream ID	2
Component Stream Identifier	m

Maximum Class 4 Stream Bandwidth limit:

This unsigned 32 bit field shall, in units of Bytes per second, identify the largest bandwidth to be allocated to carry the Component Stream.

Minimum Class 4 Stream Bandwidth limit:

This unsigned 32 bit field shall, in units of Bytes per second, identify the smallest bandwidth to be allocated to carry the Component Stream.

Stream End–to–End Credit: This unsigned 16 bit field shall identify the largest number of unacknowledged Data Frames to support for this Component Stream.

The value for this field shall not exceed $\frac{3\ \text{Latency}_{max}\ \text{Bandwidth}_{max}}{\text{Payload}_{max}}$, rounded up towards the next higher integer value, where $\text{Latency}_{\text{max}}$ = Maximum Class 4 Stream Latency, $\text{Bandwidth}_{\text{max}}$ = Maximum Class 4 Stream

Bandwidth limit and $\text{Payload}_{\text{max}}$ = Stream Data Field Size.

The value of this field shall for Class 4 Real–Time Streams exceed $\frac{2\ \text{Latency}_{max}\ \text{Bandwidth}_{min}}{\text{Payload}_{max}}$, rounded up towards the next higher integer value, where $\text{Latency}_{\text{max}}$ = Maximum Class 4 Stream Latency, $\text{Bandwidth}_{\text{min}}$ = Minimum Class 4 Stream Bandwidth limit and $\text{Payload}_{\text{max}}$ = Stream Data Field Size.

The System Resource Manager may decrease the value initially submitted by the Class 4 Stream Definer. However, this shall not cause any violation of the above requirements.

Stream Data Field Size: This unsigned 16 bit field, $\text{Payload}_{\text{max}}$, shall represent the largest Payload size, as defined by FC–PH, to be used for this Component Stream.

The following relationship shall hold true:

$$\text{Payload}_{\text{max}} \leq \begin{cases} \text{FLOGI Data Field Size} \\ \text{SC Data Field Size} \end{cases}$$

$$\text{Payload}_{\text{max}}\ \text{Mod}\ 4\ =\ 0$$

Length of Component Stream ID: This unsigned 16 bit field shall identify the number of bytes in the ‘Component Stream Identifier’ field. This shall be a value equal to an integer number of words. The value of this field shall not exceed 512.

Component Stream Identifier: This field shall in a vendor unique manner identify the specific piece of Audio–Visual material, such as a file[4], or a specific Audio–Visual source, such as a camera.

Reply Extended Link Service Sequence:

Service Reject (LS_RJT)

Signifies the rejection of the C4_Tag command

– Reject Reason code

The following Reject Reason codes, in addition to the FC–PH reason codes, are defined:

Invalid Class 4 Stream Consumer ID: No Class 4 Stream Port with the given Fibre Channel address exist.

Invalid Class 4 Stream Producer ID: No Class 4 Stream Port with the given Fibre Channel address exist.

Invalid Stream Type: The Class 4 Stream type defined is not supported by either or both the Class 4 Stream Producer and the Class 4 Stream Consumer.

Invalid Component Streams: The number of Component Streams exceed either or both the Class 4 Stream Producers and the Class 4 Stream Consumers capacity.

Excessive Component Streams: The number of Component Stream exceed the current capacity of either or both the Class 4 Stream Producers and the Class 4 Stream Consumers.

Invalid Maximum Bandwidth request: The value of the ‘Minimum Class 4 Stream Bandwidth limit’ field exceed the value of the ‘Maximum Class 4 Stream Bandwidth limit’ for the same Component Stream in one or more Component Stream specifications.

Invalid Bandwidth: Either or both the Class 4 Stream Producer and the Class 4 Stream Consumer in unable to support the requested ‘Minimum Class 4 Stream Bandwidth limit’.

Excessive Bandwidth: Either or both the Class 4 Stream Producer and the Class 4 Stream Consumer is unable to currently support the requested ‘Minimum Class 4 Stream Bandwidth limit’.

Invalid Component Stream ID: The Class 4 Stream Producer is unable to identify the Component Stream.

Invalid Stream End–to–End Credit: Either or both the Class 4 Stream Producer and the Class 4 Stream Consumer is unable to currently support the requested ‘Stream End–to–End Credit’.

Component Stream busy: One or more of the specified Component Streams are currently unavailable.

---

[4] This may specify a specific starting location, frame.

Invalid Activation Time: The time value specified by the 'Activation Time' and 'Activation Time Fraction' fields is illegal.

Invalid Termination Time: The time value specified by the 'Termination Time' and 'Termination Time Fraction' fields is illegal.

Tag unavailable: Either the Class 4 Stream Producer have no free Producer Tag or the System Resource Manager have no free Stream Tag.

Insufficient Fabric bandwidth: The Fabric is unable to support the requested Minimum Bandwidth for the Class 4 Stream.

Excessive Fabric Latency: The Fabric is unable to support a Class 4 Stream meeting the requested Maximum Class 4 Stream Latency guarantee.

Accept (ACC)

Signifies acceptance of the C4_Tag command

- Accept Payload

The format of the Accept Payload is shown in table 61.

Table 61 – C4_Tag Accept Payload

Item	Size (Bytes)
hex '02 00 00 00'	4
Class 4 Stream Pacing controller	1
Stream Concurrent Sequences	1
Tag	2

Class 4 Stream Pacing controller: This unsigned 8 bit field shall identify the entity responsible for maintaining the bandwidth limits for the Class 4 Stream.

The following values are defined for the Class 4 Stream Pacing controller field:

- *1:* Consumer Pacing;
- *2:* Producer Pacing;
- *3:* Fabric Pacing;
- *other values:* reserved.

This field shall be set = 1 (Consumer Pacing) for both Class 4 Asynchronous Streams and Class 4 Available Bit Rate Streams.

The System Resource Manager shall not override the Class 4 Stream Producers choice.

Stream Concurrent Sequences: This unsigned 8 bit field shall identify the largest number of Open Sequence, which shall be used for this Class 4 Stream.

This field shall be set = 1 for both Class 4 Asynchronous Streams and Class 4 Available Bit Rate Streams.

The System Resource Manager should not override the Class 4 Stream Producers choice.

Tag: An unsigned 16 bit field identifying the Class 4 Stream. The Class 4 Stream Producer shall return a Producer Tag and the System Resource Manager a Stream Tag.

6.2.3 Class 4 Stream circuit Control

The System Resource Manager sends the Begin Stream (C4_Circuit) Extended Link Service command to the Quality of Service Facilitator to set the Fabric parameters for a Class 4 Stream circuit.

The C4_Circuit command shall travel on Permanent Stream #1.

Protocol:

Single frame Begin Stream request Sequence
Single frame Accept reply Sequence

Format: FT_1

Addressing: The S_ID shall designate the System Resource Manager. The D_ID shall designate the Quality of Service Facilitator, at the well known address hex 'FFFFF9'.

Payload: The format of the C4_Circuit Extended Link Service command is shown in table 62.

Class 4 Stream Producer VC_ID: The VC_ID assigned to the Class 4 circuit recipient (CTR) by the System Resource Manager, i.e. the VC_ID for the virtual circuit in the direction from the Class 4 Stream Producer to the Class 4 Stream Consumer.

The value of this filed shall be limited to the range of 1 to the value of 'Max Class 4 Stream circuits' for the Class 4 Stream Producer Port as a Producer.

Table 62 – C4_Circuit Payload

Item	Size (Bytes)
hex ‘xx 00 00 00’	4
Class 4 Stream Producer VC_ID	1
Founding Stream Producer ID	3
Class 4 Stream Producer Maximum Class 4 Stream Bandwidth limit	4
Class 4 Stream Producer Minimum Class 4 Stream Bandwidth limit	4
Class 4 Stream Consumer VC_ID	1
Founding Stream Consumer ID	3
Class 4 Stream Consumer Maximum Class 4 Stream Bandwidth limit	4
Class 4 Stream Consumer Minimum Class 4 Stream Bandwidth limit	4
Live Class 4 Stream Consumer VC Credit limit	1
Live Class 4 Stream Producer VC Credit limit	1
Reserved	1
Class 4 Stream circuit type	1
ISL Class 4 Stream circuit parameter set #1	8
…	
ISL Class 4 Stream circuit parameter set #n	8

Founding Stream Producer ID: This unsigned 24 bit field shall identify the Class 4 circuit recipient (CTR) of the Class 4 Stream circuit.

Class 4 Stream Producer Maximum Class 4 Stream Bandwidth limit: This unsigned 32 bit field shall, in units of Bytes per second, identify the upper bounds on the bandwidth allocated to the Class 4 Stream circuit in the direction from the Class 4 Stream Producer (CTR) to the Class 4 Stream Consumer (CTI).

Class 4 Stream Producer Minimum Class 4 Stream Bandwidth limit: This unsigned 32 bit field shall, in units of Bytes per second, identify the lower bounds on the bandwidth allocated to the Class 4 Stream circuit in the direction from the Class 4 Stream Producer (CTR) to the Class 4 Stream Consumer (CTI).

Class 4 Stream Consumer VC_ID: The VC_ID assigned to the Class 4 circuit initiator (CTI) by the System Resource Manager, i.e. the VC_ID for the virtual circuit in the direction from the Class 4 Stream Consumer to the Class 4 Stream Producer.

The value of this filed shall be limited to the range of 1 to the value of ‘Max Class 4 Stream circuits’ for the Class 4 Stream Consumer Port as a Consumer.

Founding Stream Consumer ID: This unsigned 24 bit field shall identify the Class 4 circuit initiator (CTI) of the Class 4 Stream circuit.

Class 4 Stream Consumer Maximum Class 4 Stream Bandwidth limit: This unsigned 32 bit field shall, in units of Bytes per second, identify the upper bounds on the bandwidth allocated to the Class 4 Stream circuit in the direction from the Class 4 Stream Consumer (CTI) to the Class 4 Stream Producer (CTR).

Class 4 Stream Consumer Maximum Class 4 Stream Bandwidth limit: This unsigned 32 bit field shall, in units of Bytes per second, identify the lower bounds on the bandwidth allocated to the Class 4 Stream circuit in the direction from the Class 4 Stream Consumer (CTI) to the Class 4 Stream Producer (CTR).

Live Class 4 Stream Consumer VC Credit limit: This unsigned 8 bit field shall identify the Buffer–to–Buffer credit limit, for the Live state, assigned, by the System Resource Manager, to the Class 4 Stream circuit in the direction from the Class 4 Stream Consumer (CTI) to the Class 4 Stream Producer (CTR).

Live Class 4 Stream Producer VC Credit limit: This unsigned 8 bit field shall identify the Buffer–to–Buffer credit limit, for the Live state, assigned, by the System Resource Manager, to the Class 4 Stream circuit in the direction from the Class 4 Stream Producer (CTR) to the Class 4 Stream Consumer (CTI).

Class 4 Stream circuit type: This unsigned 8 bit field shall identify the type of Class 4 Stream circuit to establish. Four values are defined for this field:

- *0:* Conventional Class 4 circuit;
- *1:* Permanent Stream circuit;
- *2:* Simple Class 4 Stream circuit;
- *3:* Composite Class 4 Stream circuit;
- *other values:* reserved.

ISL Class 4 Stream circuit parameter set:

Zero, one or more entries for ISL Class 4 Stream circuit parameter sets are defined. The format for a ISL Class 4 Stream circuit parameter set is shown in table 63.

Table 63 – Class 4 Stream circuit – ISL Parameters

Item	Size (Bytes)
Reserved	1
Inter–Switch Link Identifier	3
Live local CTI to CTR VC Credit	1
Live local CTR to CTI VC Credit	1
Local CTI to CTR Circuit ID	1
Local CTR to CTI Circuit ID	1

Inter–Switch Link Identifier: This unsigned 24 bit field shall identify the lower valued Fibre Channel Switch Port address for the Inter–Switch Link (ISL).

Live local CTI to CTR VC Credit: This unsigned 8 bit field shall identify the Buffer–to–Buffer credit limit, for the Live state, assigned, by the System Resource Manager, to the Class 4 Stream circuit in the direction from the Class 4 Stream Consumer (CTI) to the Class 4 Stream Producer (CTR).

Live local CTR to CTI VC Credit: This unsigned 8 bit field shall identify the Buffer–to–Buffer credit limit, for the Live state, assigned, by the System Resource Manager, to the Class 4 Stream circuit in the direction from the Class 4 Stream Producer (CTR) to the Class 4 Stream Consumer (CTI).

Local CTI to CTR Circuit ID: This unsigned 8 bit field, assigned by the System Resource Manager, shall identify the Class 4 Stream circuit in the direction from the Class 4 Stream Consumer (CTI) to the Class 4 Stream Producer (CTR).

Local CTI to CTR Circuit ID: This unsigned 8 bit field, assigned by the System Resource Manager, shall identify the Class 4 Stream circuit in the direction from the Class 4 Stream Producer (CTR) to the Class 4 Stream Consumer (CTI).

Reply Extended Link Service Sequence:

Service Reject (LS_RJT)

Signifies the rejection of the C4_Circuit command

- Reject Reason code

The following Reject Reason codes, in addition to the FC–PH reason codes, are defined:

Invalid Founding Stream Producer ID: No Class 4 Stream Port with the given Fibre Channel address exist.

Invalid Class 4 Stream Producer VC_ID: The identified Class 4 Stream Producer VC_ID can not be assigned to a Class 4 Stream circuit.

Invalid Founding Stream Consumer ID: No Class 4 Stream Port with the given Fibre Channel address exist.

Invalid Class 4 Stream Consumer VC_ID:

The identified Class 4 Stream Consumer VC_ID can not assigned to a Class 4 Stream circuit.

Unable to establish Class 4 Stream circuit:

Establishment of the Class 4 Stream circuit is impossible at the current time, because of lack of resources, by either the:

- F_Port attached to the Class 4 Stream Consumer;
- F_Port attached to the Class 4 Stream Producer
- Fabric;

Accept (ACC)
Signifies acceptance of the C4_Circuit command

- Accept Payload

The format of the Accept Payload is shown in table 64.

Table 64 – C4_Circuit Accept Payload

Item	Size (Bytes)
hex ‘02 00 00 00’	4

6.2.4 Class 4 Stream circuit End

The Quality of Service Facilitator sends a Class 4 Stream circuit End (C4_End) Extended Link Service command to the System Resource Manager, informing it of the removal of a Class 4 Stream circuit.

The C4_End command shall be sent on Permanent Stream #1.

Protocol:

Single frame Class 4 Stream circuit End request Sequence
Single frame Accept reply Sequence

Format: FT_1

Addressing: The D_ID shall designate the System Resource Manager. The S_ID shall designate the Quality of Service Facilitator.

Payload: The format of the C4_End Extended Link Service command is shown in table 65.

Founding Stream Producer VC_ID: This unsigned 8 bit field shall indicate the Class 4 virtual circuit identifier, assigned by the System Resource Manager, to the virtual circuit in the direction from the Founding Stream Producer (CTR) to the Founding Stream Consumer (CTI).

Founding Stream Producer ID: This unsigned 24 bit field shall indicate the Fibre Channel address of the Founding Stream Producer (CTR).

Table 65 – C4_End Payload

Item	Size (Bytes)
hex ‘xx 00 00 00’	4
Founding Stream Producer VC_ID	1
Founding Stream Producer ID	3
Founding Stream Consumer VC_ID	1
Founding Stream Consumer ID	3

Founding Stream Consumer VC_ID: This unsigned 8 bit field shall indicate the Class 4 virtual circuit identifier, assigned by the System Resource Manager, for the virtual circuit in the direction from the Founding Stream Consumer (CTI) to the Founding Stream Producer (CTR).

Founding Stream Consumer ID: This unsigned 24 bit field shall indicate the Fibre Channel address of the Founding Stream Consumer (CTI).

Reply Extended Link Service Sequence:

Service Reject (LS_RJT)
Signifies the rejection of the Query_PT command

- Reject Reason code

The following Reject Reason codes, in addition to the FC–PH reason codes, are defined, presented in order of priority:

Invalid Founding Stream Producer ID: The Founding Stream Producer ID is not recognized by the System Resource Manager, i.e. the System Resource Manager is unaware of any Class 4 Stream circuit established to the given Founding Stream Producer ID.

Invalid Founding Stream Producer VC_ID:

The Founding Stream Producer VC_ID is not recognized by the System Resource Manager, i.e. the System Resource Manager is unaware of any Class 4 Stream circuit from the Founding Stream Producer with the given Founding Stream Producer VC_ID.

Invalid Founding Stream Consumer ID: The Founding Stream Consumer ID is not recognized by the System Resource Manager, i.e.

the System Resource Manager is unaware of any Class 4 Stream circuit established to the given Founding Stream Consumer ID.

Invalid Founding Stream Consumer VC_ID:

The Founding Stream Consumer VC_ID is not recognized by the System Resource Manager, i.e. the System Resource Manager is unaware of any Class 4 Stream circuit at the Founding Stream Consumer with the given Founding Stream Consumer VC_ID.

Accept (ACC)
Signifies acceptance of the C4_End command

– Accept Payload

The format of the Accept Payload is shown in table 66.

Table 66 – C4_End Accept Payload

Item	Size (Bytes)
hex ‘02 00 00 00’	4

6.2.5 Class 4 Stream Notify

The System Resource Manager sends the Class 4 Stream Notify (C4_Notify) Extended Link Service command to a Class 4 Stream Consumer informing it of commitments made to a new Class 4 Stream by the System Resource Manager on the behalf of the Class 4 Stream Consumer.

The C4_Notify command shall travel on Permanent Stream #2.

Protocol:

Single frame Class 4 Stream Notify request Sequence
Single frame Accept reply Sequence

Format: FT_1

Addressing: The S_ID shall designate the System Resource Manager. The D_ID shall designate the Class 4 Stream Consumer.

Payload: The format of the C4_Notify Extended Link Service command is shown in table 67.

Table 67 – C4_Notify Payload

Item	Size (Bytes)
hex ‘xx 00 00 00’	4
Activation Time	4
Activation Time Fraction	4
Termination Time	4
Termination Time Fraction	4
Class 4 Stream Producer VC_ID	1
Class 4 Stream Producer ID	3
Class 4 Stream Consumer VC_ID	1
Class 4 Stream Consumer ID	3
Maximum Class 4 Stream Latency	2
Reserved	2
Producer Tag	2
Stream Tag	2
Stream Type	1
Class 4 Stream Pacing controller	1
Stream Concurrent Sequences	1
Number of Component Streams	1
Component Stream parameter set #1	12
...	
Component Stream parameter set #n	12

Activation Time: This unsigned 32 bit field shall identify the time in seconds, relative to 00:00:00 Universal Time (UT) on 1 January 1900, at which transmission for the Class 4 Stream will begin.

Activation Time Fraction: This unsigned 32 bit field shall identify an offset, in units of 1/4 294 967 297 second, to the Activation Time.

An immediate start time, within 1/32 second, for the Class 4 Stream is specified, if both the ‘Activation Time’ and the ‘Activation Time Fraction’ fields are set = hex ‘FF FF FF FF’.

The Activation Time shall not be set so that both the ‘Activation Time’ and the ‘Activation Time Fraction’ fields are = hex ‘00 00 00 00’.

Termination Time: This unsigned 32 bit field shall identify the time in seconds, relative to 00:00:00 Universal Time (UT) on 1 January 1900, at which transmission for the identified Class 4 Stream will cease.

Termination Time Fraction: This unsigned 32 bit field shall identify an offset, in units of 1/4 294 967 297 second, to the Termination Time.

No Termination Time for the Class 4 Stream is specified, if both the ‘Termination Time’ and the ‘Termination Time Fraction’ fields are set = hex ‘00 00 00 00’.

The Termination Time shall not be set so that both the ‘Termination Time’ and the ‘Termination Time Fraction’ fields are = hex ‘FF FF FF FF’ (immediate termination).

Class 4 Stream Producer VC_ID: This unsigned 8 bit field shall indicate the virtual circuit identifier assigned, by the System Resource Manager, to the virtual circuit in the direction from the Class 4 Stream Producer to the Class 4 Stream Consumer.

Class 4 Stream Producer ID: This unsigned 24 bit field shall indicate the source of the Class 4 Stream.

Class 4 Stream Consumer VC_ID: This unsigned 8 bit field shall indicate the virtual circuit identifier assigned, by the System Resource Manager, to the virtual circuit in the direction from the Class 4 Stream Consumer to the Class 4 Stream Producer.

Class 4 Stream Consumer ID: This unsigned 24 bit field shall indicate the destination of the Class 4 Stream.

Maximum Class 4 Stream Latency: This unsigned 16 bit field shall indicate the longest time, in units of 1/65 536 second, that a Class 4 frame shall spend in the Fabric.

Producer Tag: This unsigned 16 bit field, assigned by the Class 4 Stream Producer, shall in conjunction with the Class 4 Stream Producer ID, uniquely identify this Class 4 Stream.

Stream Tag: This unsigned 16 bit field, assigned by the System Resource Manager, shall in conjunction with the Class 4 Stream Producer ID, uniquely identify this Class 4 Stream.

Stream Type: This unsigned 8 bit field shall identify the Stream Type to be used by this Class 4 Stream.

The following values are defined for the Stream Type field:

- *1:* Class 4 Asynchronous Stream;
- *2:* Class 4 Available Bit Rate Stream;
- *3:* Class 4 Real–Time Stream;
- *other values:* reserved.

Class 4 Stream Pacing controller: This unsigned 8 bit field shall identify the Class 4 Stream Pacing controller method used by this Class 4 Stream.

The following values are defined for the Class 4 Stream Pacing controller field:

- *1:* Consumer Pacing;
- *2:* Producer Pacing;
- *3:* Fabric Pacing;
- *other values:* reserved.

Stream Concurrent Sequences: This unsigned 8 bit field shall identify the largest number of Open Sequence, to be used by this Class 4 Stream.

Number of Component Streams: This unsigned 8 bit field shall identify the number of Component Streams used by the Class 4 Stream.

Component Stream parameter set: One or more entries for Component Stream parameters are defined, as specified by the ‘Number of Component Streams’ field. The format of a Component Stream parameter set is shown in table 68.

Maximum Class 4 Stream Bandwidth limit:

This unsigned 32 bit field shall, in units of Bytes per second, identify the largest bandwidth allocated to carry the Component Stream.

Table 68 – Component Stream – Parameters

Item	Size (Bytes)
Maximum Class 4 Stream Bandwidth limit	4
Minimum Class 4 Stream Bandwidth limit	4
Stream End–to–End Credit	2
Stream Data Field Size	2

Minimum Class 4 Stream Bandwidth limit:

This unsigned 32 bit field shall, in units of Bytes per second, identify the smallest bandwidth allocated to carry the Component Stream.

Stream End–to–End Credit: This unsigned 16 bit field shall identify the largest number of unacknowledged Data Frames to support for this Component Stream.

Stream Data Field Size: This unsigned 16 bit field shall represent the largest Payload size, as defined by FC–PH, used by this Component Stream.

Reply Extended Link Service Sequence:

Service Reject (LS_RJT)
Signifies the rejection of the C4_Notify command

– Reject Reason code

The following Reject Reason codes, in addition to the FC–PH reason codes, are defined:

Invalid Class 4 Stream Consumer ID: The value of the Class 4 Stream Consumer ID field does not match the D_ID value.

Invalid Stream Type: The Class 4 Stream type defined is not supported by the Class 4 Stream Consumer.

Invalid Component Streams: The number of Component Streams exceed the Class 4 Stream Consumers capacity.

Invalid Bandwidth: The Class 4 Stream Consumer in unable to support the requested ‘Minimum Class 4 Stream Bandwidth limit’.

Invalid Activation Time: The time value specified by the ‘Activation Time’ and ‘Activation Time Fraction’ fields is illegal.

Invalid Termination Time: The time value specified by the ‘Termination Time’ and ‘Termination Time Fraction’ fields is illegal.

Accept (ACC)
Signifies acceptance of the C4_Notify command

– Accept Payload

The format of the Accept Payload is shown in table 69.

Table 69 – C4_Notify Accept Payload

Item	Size (Bytes)
hex ‘02 00 00 00’	4

6.2.6 Get Class 4 Stream Ability

Class 4 Stream Ports sends, on Permanent Stream #2, a Get Class 4 Stream Ability (C4_Ability) Extended Link Service command to the System Resource Manager, to obtain the Class 4 Stream capabilities of another Class 4 Stream Port.

The System Resource Manager sends, on Permanent Stream #2, a Get Class 4 Stream Ability Extended Link Service command to a Class 4 Stream Port, whose Class 4 Stream capabilities it wish to obtain.

The System Resource Manager sends, on Permanent Stream #1, a Get Class 4 Stream Ability Extended Link Service command to the Quality of Service Facilitator to obtain the Class 4 Stream capabilities of a Switch Port (F_Port or E_Port).

Protocol:

Get Class 4 Stream Ability request Sequence
Accept reply Sequence

Format: FT_1

Addressing: The S_ID field designates the source, wishing to obtain another Ports Class 4 Stream capabilities.

The D_ID field designates the System Resource Manager for a general inquiry on another Ports Class 4 Stream capabilities.

The D_ID field designated the Class 4 Stream Port whose Class 4 Stream capabilities the System Resource Manager wish to obtain, for System Resource Manager initiated requests.

The D_ID field designated the Quality of Service Facilitator, at the well known address hex ‘FFFFF9’, when the System Resource Manager wish to obtain the Class 4 Stream capabilities of a Switch Port.

Payload: The format of the C4_Ability Extended Link Service command is shown in table 70.

Table 70 – C4_Ability Payload

Item	Size (Bytes)
hex ‘xx 00 00 00’	4
Reserved	1
Class 4 Stream Port ID	3

Class 4 Stream Port ID: The Fibre Channel address for the Class 4 Stream Port whose Class 4 Stream capabilities are desired.

Reply Extended Link Service Sequence:

Service Reject (LS_RJT)

Signifies the rejection of the C4_Ability command

– Reject Reason code

The following Reject Reason codes, in addition to the FC–PH reason codes, are defined:

Invalid Class 4 Stream Port ID: No Class 4 Stream Port with the given Fibre Channel address exist.

Accept (ACC)

Signifies acceptance of the C4_Ability command

– Accept Payload

The format of the Accept Payload, for a C4_Ability command sent to the System Resource Manager for an N_Port, is shown in table 71. The format of the Accept Payload, for a C4_Ability command sent to the Quality of Service Facilitator, is shown in table 72.

Table 71 – C4_Ability Accept Payload – N_Port

Item	Size (Bytes)
hex ‘02 00 00 00’	4
Class 4 Stream Producer Abilities	48
Class 4 Stream Consumer Abilities	68

Table 72 – C4_Ability Accept Payload – Switch Port

Item	Size (Bytes)
hex ‘02 00 00 00’	4
Reserved	1
Attached Port ID	3
Max. Class 4 Stream Bandwidth	4
Max. Class 4 Stream circuit Bandwidth	4
Min. Class 4 Stream circuit Bandwidth	4
Class 4 Stream circuit Bandwidth quantum	4
Max. number of Class 4 Stream circuits	1
Max. Live Credit per Class 4 Stream circuit	1
Max. Live Credit for Class 4 Streams	2

Class 4 Stream Producer Abilities: The format of the Ports Class 4 Stream abilities as a Stream Producer are shown in table 73.

Class 4 Stream Consumer Abilities: The format of the Ports Class 4 Stream abilities as a Stream Consumer are shown in table 74.

Attached Port ID: This unsigned 24 bit field shall identify the Fibre Channel address of the Port attached to the Class 4 Stream Port whose Class 4 Stream capabilities are sought.

Table 73 – Class 4 Stream Producer Abilities

Item	Size (Bytes)
hex ‘02 00 00 00’	4
Max. Class 4 Streams	2
Max. Class 4 Asynchronous Streams	2
Max. Class 4 Available Bit Rate Streams	2
Max. Class 4 Real–Time Streams	2
Max. self pacing Class 4 Real–Time Streams	2
Max Class 4 Stream circuits	1
Max. Simple Class 4 Stream circuits	1
Max. Composite Class 4 Stream circuits	1
Max. Class 4 Streams per Composite Class 4 Stream circuit	1
Max. Class 4 Asynchronous Streams per Composite Class 4 Stream circuit	1
Max. Class 4 ABR Streams per Composite Class 4 Stream circuit	1
Max. Class 4 RT Streams per Composite Class 4 Stream circuit	1
Reserved	1
Max. Component Streams per Class 4 Available Bit Rate Stream	1
Max. Component Streams per Class 4 Real–Time Stream	1
Max. Bandwidth	4
Max. aggregate Bandwidth for Available Bit Rate Streams	4
Max. aggregate Bandwidth for Real–Time Streams	4
Max. Bandwidth per Class 4 Asynchronous Stream	4
Max. Bandwidth per Class 4 Available Bit Rate Stream	4
Max. Bandwidth per Class 4 Real–Time Stream	4

(continued)

Table 73 (concluded)

This shall be the address of the either an N_Port, if the Class 4 Stream capabilities of an F_Port are sought, or an E_Port, if the Class 4 Stream capabilities of an E_Port are sought.

Max. Class 4 Stream Bandwidth: This unsigned 32 bit field shall identify the highest Bandwidth, in Bytes per second, this Switch Port is able to accept, receive, across all Class 4 Streams.

Max. Class 4 Stream circuit Bandwidth: This unsigned 32 bit field shall identify the highest Bandwidth, in Bytes per second, this Switch Port is able to accept, receive, for a single Class 4 Stream circuit.

Min. Class 4 Stream circuit Bandwidth: This unsigned 32 bit field shall identify the smallest Bandwidth, in Bytes per second, this Switch Port is able to allocate for reception on a single Class 4 Stream circuit.

Class 4 Stream circuit Bandwidth quantum:

This unsigned 32 bit field shall identify the smallest amount of Bandwidth increment, in Bytes per second, which this Switch Port is able to allocate for reception on a single Class 4 Stream circuit.

Max. number of Class 4 Stream circuits:

This unsigned 8 bit field shall identify the largest number of Class 4 Stream circuits this Switch Port is able to accept, receive.

Max. Live Credit per Class 4 Stream circuit:

This unsigned 8 bit field shall identify the largest number of Buffer–to–Buffer Credits, this Switch

Table 74 – Class 4 Stream Consumer Abilities

Item	Size (Bytes)
Max. Class 4 Streams	2
Max. Class 4 Asynchronous Streams	2
Max. Class 4 Available Bit Rate Streams	2
Max. Class 4 Real–Time Streams	2
Max. pacing Class 4 Real–Time Streams	2
Reserved	3
Max Class 4 Stream circuits	1
Max. Simple Class 4 Stream circuits	1
Max. Composite Class 4 Stream circuits	1
Max. Class 4 Streams per Composite Class 4 Stream circuit	1
Max. Class 4 Asynchronous Streams per Composite Class 4 Stream circuit	1
Max. Class 4 Available Bit Rate Streams per Composite Class 4 Stream circuit	1
Max. Class 4 Real–Time Streams per Composite Class 4 Stream circuit	1
Max. Bandwidth	4
Max. aggregate Bandwidth for Available Bit Rate Streams	4
Max. aggregate Bandwidth for Real–Time Streams	4
Max. Bandwidth per Class 4 Asynchronous Stream	4
Max. Bandwidth per Class 4 Available Bit Rate Stream	4
Max. Bandwidth per Class 4 Real–Time Stream	4
Reserved	2
Max. RX_ID assignment latency for Class 4 Asynchronous Streams	2
Max latency for initial ACK for Class 4 Asynchronous Streams	2
Max latency for final ACK for Class 4 Asynchronous Streams	2
Reserved	2
Max. RX_ID assignment latency for Class 4 Available Bit Rate Streams	2
Max latency for initial ACK for Class 4 Available Bit Rate Streams	2
Max latency for final ACK for Class 4 Available Bit Rate Streams	2
Reserved	2
Max. RX_ID assignment latency for Class 4 Real–Time Streams	2
Max latency for initial ACK for Class 4 Real–Time Streams	2
Max latency for final ACK for Class 4 Real–Time Streams	2

(continued)

Table 74 (concluded)

Port is able to allocate to a single Class 4 Stream circuit.

Max. Live Credit for Class 4 Streams: This unsigned 16 bit field shall identify the largest number of Buffer–to–Buffer Credits, this Switch Port is able to allocate across all Class 4 Stream circuits.

6.2.6.1 Class 4 Stream Producer Abilities

Max. Class 4 Streams: This unsigned 16 bit field, C4 Stream$_{max}$, shall identify the largest number of Class 4 Streams a Stream Producer is able to support.

The following relationship shall hold true:

$$\text{C4 Stream}_{\text{max}} \geq \begin{cases} \text{C4 Async Stream}_{\text{max}} \\ \text{C4 ABR Stream}_{\text{max}} \\ \text{C4 RT Stream}_{\text{max}} \end{cases}$$

Max. Class 4 Asynchronous Streams: This unsigned 16 bit field, $\text{C4 Async Stream}_{\text{max}}$, shall identify the largest number of Class 4 Asynchronous Streams a Stream Producer is able to support.

Max. Class 4 Available Bit Rate Streams:

This unsigned 16 bit field, $\text{C4 ABR Stream}_{\text{max}}$, shall identify the largest number of Class 4 Available Bit Rate Streams a Stream Producer is able to support.

Max. Class 4 Real–Time Streams: This unsigned 16 bit field, $\text{C4 RT Stream}_{\text{max}}$, shall identify the largest number of Class 4 Real–Time Streams a Stream Producer is able to support.

Max. self pacing Class 4 Real–Time Streams: This unsigned 16 bit field shall identify the largest number of Class 4 Real–Time Streams on which a Stream Producer is able to support Producer Pacing.

Max Class 4 Stream circuits: This unsigned 8 bit field, $\text{C4 circuits}_{\text{max}}$, shall identify the largest number of Class 4 Stream circuits a Stream Producer is able to support[5] .

The following relationship shall hold true:

$$\text{C4 circuits}_{\text{max}} \geq \begin{cases} \text{C4 Simple circuits}_{\text{max}} \\ \text{C4 Composite circuits}_{\text{max}} \end{cases}$$

Max. Simple Class 4 Stream circuits: This unsigned 8 bit field, $\text{C4 Simple circuits}_{\text{max}}$, shall identify the largest number of Simple Class 4 Stream circuits a Stream Producer is able to support.

Max. Composite Class 4 Stream circuits:

This unsigned 8 bit field, $\text{C4 Composite circuits}_{\text{max}}$, shall identify the largest number of Composite Class 4 Stream circuits a Stream Producer is able to support.

Max. Class 4 Streams per Composite Class 4 Stream circuit: This unsigned 8 bit field, $\text{Comp Stream}_{\text{max}}$, shall identify the largest number of Class 4 Streams a Stream Producer is able to support in a single Composite Class 4 Stream circuit.

The following relationship shall hold true:

$$\text{Comp Stream}_{\text{max}} \geq \begin{cases} \text{Comp Async Stream}_{\text{max}} \\ \text{Comp ABR Stream}_{\text{max}} \\ \text{Comp RT Stream}_{\text{max}} \end{cases}$$

Max. Class 4 Asynchronous Streams per Composite Class 4 Stream circuit: This unsigned 8 bit field, $\text{Comp Async Stream}_{\text{max}}$, shall identify the largest number of Class 4 Asynchronous Streams a Stream Producer is able to support in a single Composite Class 4 Stream circuit.

Max. Class 4 ABR Streams per Composite Class 4 Stream circuit: This unsigned 8 bit field, $\text{Comp ABR Stream}_{\text{max}}$, shall identify the largest number of Class 4 Available Bit Rate Streams a Stream Producer is able to support in a single Composite Class 4 Stream circuit.

Max. Class 4 RT Streams per Composite Class 4 Stream circuit: This unsigned 8 bit field, $\text{Comp RT Stream}_{\text{max}}$, shall identify the largest number of Class 4 Real–Time Streams a Stream Producer is able to support in a single Composite Class 4 Stream circuit.

Max. Component Streams per Class 4 Available Bit Rate Stream: This unsigned 8 bit field shall identify the largest number of Component Streams per Class 4 Available Bit Rate Stream a Stream Producer is able to support.

Max. Component Streams per Class 4 Real–Time Stream: This unsigned 8 bit field shall identify the largest number of Component Streams per Class 4 Real–Time Stream a Stream Producer is able to support.

---

[5] The total number of Class 4 Stream circuits a Class 4 Stream Port is able to support is equal to the larger of the ‘Max Class 4 Stream circuits’ fields of either the Class 4 Stream Producer Abilities or the Class 4 Stream Consumer Abilities.

Max. Bandwidth: This unsigned 32 bit field, $\text{Bandwidth}_{\text{max}}$, shall identify the highest bandwidth, in units of Bytes per second, that a Stream Producer is able to support across all Class 4 Streams.

The following relationship shall hold true:

$$\text{Bandwidth}_{\text{max}} \geq \begin{cases} \text{Async Bandwidth}_{\text{max}} \\ \text{Total ABR Bandwidth}_{\text{max}} \\ \text{Total RT Bandwidth}_{\text{max}} \end{cases}$$

Max. aggregate Bandwidth for Available Bit Rate Streams: This unsigned 32 bit field, $\text{Total ABR Bandwidth}_{\text{max}}$, shall identify the highest bandwidth, in units of Bytes per second, that a Stream Producer is able to support across all Available Bit Rate Streams.

The following relationship shall hold true:

$$\text{Total ABR Bandwidth}_{\text{max}} \geq \text{ABR Bandwidth}_{\text{max}}$$

Max. aggregate Bandwidth for Real–Time Streams: This unsigned 32 bit field, $\text{Total RT Bandwidth}_{\text{max}}$, shall identify the highest bandwidth, in units of Bytes per second, that a Stream Producer is able to support across all Real–Time Streams.

The following relationship shall hold true:

$$\text{Total RT Bandwidth}_{\text{max}} \geq \text{RT Bandwidth}_{\text{max}}$$

Max. Bandwidth per Class 4 Asynchronous Stream: This unsigned 32 bit field, $\text{Async Bandwidth}_{\text{max}}$, shall identify the highest bandwidth, in units of Bytes per second, that a Stream Producer is able to support for a single Asynchronous Stream.

Max. Bandwidth per Class 4 Available Bit Rate Stream: This unsigned 32 bit field, $\text{ABR Bandwidth}_{\text{max}}$, shall identify the highest bandwidth, in units of Bytes per second, that a Stream Producer is able to support for a single Available Bit Rate Stream.

Max. Bandwidth per Class 4 Real–Time Stream: This unsigned 32 bit field, $\text{RT Bandwidth}_{\text{max}}$, shall identify the highest bandwidth, in units of Bytes per second, that a Stream Producer is able to support for a single Real–Time Stream.

6.2.6.2 Class 4 Stream Consumer Abilities

Max. Class 4 Streams: This unsigned 16 bit field, $\text{C4 Stream}_{\text{Max}}$, shall identify the largest number of Class 4 Streams a Stream Consumer is able to accept.

The following relationship shall hold true:

$$\text{C4 Stream}_{\text{Max}} \geq \begin{cases} \text{C4 Async Stream}_{\text{Max}} \\ \text{C4 ABR Stream}_{\text{Max}} \\ \text{C4 RT Stream}_{\text{Max}} \end{cases}$$

Max. Class 4 Asynchronous Streams: This unsigned 16 bit field, $\text{C4 Async Stream}_{\text{Max}}$, shall identify the largest number of Class 4 Asynchronous Streams a Stream Consumer is able to accept.

Max. Class 4 Available Bit Rate Streams: This unsigned 16 bit field, $\text{C4 ABR Stream}_{\text{Max}}$, shall identify the largest number of Class 4 Available Bit Rate Streams a Stream Consumer is able to accept.

Max. Class 4 Real–Time Streams: This unsigned 16 bit field, $\text{C4 RT Stream}_{\text{Max}}$, shall identify the largest number of Class 4 Real–Time Streams a Stream Consumer is able to accept.

Max. pacing Class 4 Real–Time Streams: This unsigned 16 bit field shall identify the largest number of Class 4 Real–Time Streams on which a Stream Consumer is able to support pacing.

Max Class 4 Stream circuits: This unsigned 8 bit field, $\text{C4 circuits}_{\text{Max}}$, shall identify the largest number of Class 4 Stream circuits a Stream Consumer is able to accept[5)].

The following relationship shall hold true:

$$\text{C4 circuits}_{\text{Max}} \geq \begin{cases} \text{C4 Simple circuits}_{\text{Max}} \\ \text{C4 Composite circuits}_{\text{Max}} \end{cases}$$

Max. Simple Class 4 Stream circuits: This unsigned 8 bit field, $\text{C4 Simple circuits}_{\text{Max}}$, shall identify the largest number of Simple Class 4 Stream circuits a Stream Consumer is able to accept.

Max. Composite Class 4 Stream circuits:

This unsigned 8 bit field, $\text{C4 Composite circuits}_{\text{Max}}$, shall identify the largest number of Class 4 Stream circuits a Stream Consumer is able to accept.

Max. Class 4 Streams per Composite Class 4 Stream circuit: This unsigned 8 bit field, $\text{Comp Stream}_{\text{Max}}$, shall identify the largest number of Class 4 Streams a Stream Consumer is able to accept in a single Composite Class 4 Stream circuit.

The following relationship shall hold true:

$$\text{Comp Stream}_{\text{Max}} \geq \begin{cases} \text{Comp Async Stream}_{\text{Max}} \\ \text{Comp ABR Stream}_{\text{Max}} \\ \text{Comp RT Stream}_{\text{Max}} \end{cases}$$

Max. Class 4 Asynchronous Streams per Composite Class 4 Stream circuit: This unsigned 8 bit field, $\text{Comp Async Stream}_{\text{Max}}$, shall identify the largest number of Class 4 Asynchronous Streams a Stream Consumer is able to accept in a single Composite Class 4 Stream circuit.

Max. Class 4 Available Bit Rate Streams per Composite Class 4 Stream circuit: This unsigned 8 bit field, $\text{Comp ABR Stream}_{\text{Max}}$, shall identify the largest number of Class 4 Available Bit Rate Streams a Stream Consumer is able to accept in a single Composite Class 4 Stream circuit.

Max. Class 4 Real–Time Streams per Composite Class 4 Stream circuit: This unsigned 8 bit field, $\text{Comp RT Stream}_{\text{Max}}$, shall identify the largest number of Class 4 Real–Time Streams a Stream Consumer is able to accept in a single Composite Class 4 Stream circuit.

Max. Bandwidth: This unsigned 32 bit field, $\text{Bandwidth}_{\text{Max}}$, shall identify the highest bandwidth, in units of Bytes per second, that a Stream Consumer is able to accept across all Class 4 Streams.

The following relationship shall hold true:

$$\text{Bandwidth}_{\text{Max}} \geq \begin{cases} \text{Async Bandwidth}_{\text{Max}} \\ \text{Total ABR Bandwidth}_{\text{Max}} \\ \text{Total RT Bandwidth}_{\text{Max}} \end{cases}$$

Max. aggregate Bandwidth for Available Bit Rate Streams: This unsigned 32 bit field, $\text{Total ABR Bandwidth}_{\text{Max}}$, shall identify the highest bandwidth, in units of Bytes per second, that a Stream Consumer is able to accept across all Available Bit Rate Streams.

The following relationship shall hold true: $\text{Total ABR Bandwidth}_{\text{Max}} \geq \text{ABR Bandwidth}_{\text{Max}}$

Max. aggregate Bandwidth for Real–Time Streams: This unsigned 32 bit field, $\text{Total RT Bandwidth}_{\text{Max}}$, shall identify the highest bandwidth, in units of Bytes per second, that a Stream Consumer is able to accept across all Real–Time Streams.

The following relationship shall hold true: $\text{Total RT Bandwidth}_{\text{Max}} \geq \text{RT Bandwidth}_{\text{Max}}$

Max. Bandwidth per Class 4 Asynchronous Stream: This unsigned 32 bit field, $\text{Async Bandwidth}_{\text{Max}}$, shall identify the highest bandwidth, in units of Bytes per second, that a Stream Consumer is able to accept for a single Asynchronous Stream.

Max. Bandwidth per Class 4 Available Bit Rate Stream: This unsigned 32 bit field, $\text{ABR Bandwidth}_{\text{Max}}$, shall identify the highest bandwidth, in units of Bytes per second, that a Stream Consumer is able to accept for a single Available Bit Rate Stream.

Max. Bandwidth per Class 4 Real–Time Stream: This unsigned 32 bit field, $\text{RT Bandwidth}_{\text{Max}}$, shall identify the highest bandwidth, in units of Bytes per second, that a Stream Consumer is able to accept for a single Real–Time Stream.

Max. RX_ID assignment latency for Class 4 Asynchronous Streams: This unsigned 16 bit field shall indicate, in units of 1/65 536 of a second, the longest time a Stream Consumer will take before assigning an RX_ID to a new Exchange, for a Data Frame received on a Class 4 Asynchronous Stream. The RX_ID is considered assigned, when the ACK frame carrying the RX_ID is transmitted by the Stream Consumer.

Max latency for initial ACK for Class 4 Asynchronous Streams: This unsigned 16 bit field shall indicate, in units of 1/65 536 of a second, the longest time a Stream Consumer will take before transmitting the ACK frame for the first Data Frame received on a new Sequence in a Class 4 Asynchronous Stream.

Max latency for final ACK for Class 4 Asynchronous Streams: This unsigned 16 bit field shall indicate, in units of 1/65 536 of a second, the longest time a Stream Consumer will take before transmitting the ACK frame for the last Data Frame received for a Sequence in a Class 4 Asynchronous Stream.

Max. RX_ID assignment latency for Class 4 Available Bit Rate Streams: This unsigned 16 bit field shall indicate, in units of 1/65 536 of a second, the longest time a Stream Consumer will take before assigning an RX_ID to a new Exchange, for a Data Frame received on a Class 4 Available Bit Rate Stream. The RX_ID is considered assigned, when the ACK frame carrying the RX_ID is transmitted by the Stream Consumer.

Max latency for initial ACK for Class 4 Available Bit Rate Streams: This unsigned 16 bit field shall indicate, in units of 1/65 536 of a second, the longest time a Stream Consumer will take before transmitting the ACK frame for the first Data Frame received on a new Sequence in a Class 4 Available Bit Rate Stream.

Max latency for final ACK for Class 4 Available Bit Rate Streams: This unsigned 16 bit field shall indicate, in units of 1/65 536 of a second, the longest time a Stream Consumer will take before transmitting the ACK frame for the last Data Frame received for a Sequence in a Class 4 Available Bit Rate Stream.

Max. RX_ID assignment latency for Class 4 Real–Time Streams: This unsigned 16 bit field shall indicate, in units of 1/65 536 of a second, the longest time a Stream Consumer will take before assigning an RX_ID to a new Exchange, for a Data Frame received on a Class 4 Real–Time Stream. The RX_ID is considered assigned, when the ACK frame carrying the RX_ID is transmitted by the Stream Consumer.

Max latency for initial ACK for Class 4 Real–Time Streams: This unsigned 16 bit field shall indicate, in units of 1/65 536 of a second, the longest time a Stream Consumer will take before transmitting the ACK frame for the first Data Frame received on a new Sequence in a Class 4 Real–Time Stream.

Max latency for final ACK for Class 4 Real–Time Streams: This unsigned 16 bit field shall indicate, in units of 1/65 536 of a second, the longest time a Stream Consumer will take before transmitting the ACK frame for the last Data Frame received for a Sequence in a Class 4 Real–Time Stream.

6.2.7 Query Tag

Class 4 Stream Definers and the System Resource Manager sends the Query Tag Extended Link Service (Query_Tag) command to obtain a list of the Tags associated with a given Class 4 Stream Producer.

The Query_Tag command shall travel on Permanent Stream #2.

Protocol:

Query Tag request Sequence
Accept reply Sequence

Format: FT_1

Addressing: The S_ID shall designate a Class 4 Stream Definer or the System Resource Manager. The D_ID shall designate the System Resource Manager, if the Query_Tag command is sent by a Class 4 Stream Definer. If the Query_Tag command is sent by the System Resource Manager, then the D_ID shall designate the Class 4 Stream Producer.

Payload: The format of the Query_Tag Extended Link Service command is shown in table 75.

Table 75 – Query_Tag Payload

Item	Size (Bytes)
hex ‘xx 00 00 00’	4
reserved	1
Class 4 Stream Producer ID	3

Class 4 Stream Producer ID: This 24 bit unsigned field shall identify the Class 4 Stream Producer for which assigned Tags are sought.

Reply Extended Link Service Sequence:

Service Reject (LS_RJT)
Signifies the rejection of the Query_Tag command
Accept (ACC)
Signifies acceptance of the Query_Tag command

– Accept Payload

The format of the Accept Payload is shown in table 76.

Table 76 – Query_Tag Accept Payload

Item	Size (Bytes)
hex ‘02 00 00 00’	4
Reserved	2
Number of assigned Tags	2
Tag data set #1	4
...	
Tag data set #n	4

Number of assigned Tags: This unsigned 16 bit field shall identify the number of Tags in use for the Stream Producer.

Tag data: Zero, one or more sets of Tag data are defined, as declared by the ‘Number of assigned Tags’ field. The format of the Tag data is shown in table 77.

Table 77 – Tag data

Item	Size (Bytes)
Stream State	1
Stream Type	1
Tag	2

Stream State: This unsigned 8 bit field shall identify the state of the Class 4 Stream.

The following states are defined for the Stream State field:

- *0:* Terminated;
- *1:* Primed;
- *2:* Dormant;
- *3:* Live;
- *All other values:* reserved.

Stream Type: This unsigned 8 bit field shall identify the Class 4 Stream Type selected for this Class 4 Stream.

Three values are defined for the Stream Type variable;

- *1:* for Class 4 Asynchronous Streams;
- *2:* for Class 4 Available Bit Rate Streams;
- *3:* for Class 4 Real–Time Streams;
- *All other values:* reserved.

Tag: This unsigned 16 bit field shall identify a specific Class 4 Stream.

The Tag field shall identify a Producer Tag if the Accept is sent by the Class 4 Stream Producer.

The Tag field shall identify a Stream Tag if the Accept is sent by the System Resource Manager.

6.2.8 Register System Resource Manager

Class 4 Stream Ports sends a Register System Resource Manager (RSRM) Extended Link Service command to the Quality of Service Facilitator (QoSF), located at the well-known address hex ‘FFFFF9’, to register their ability to become the System Resource Manager.

The RSRM command shall travel on Permanent Stream #1.

Protocol:

Register System Resource Manager request Sequence
Accept reply Sequence

Format: FT_1

Addressing: The S_ID field designates the source Port wishing to become the System Resource Manager. The D_ID designates the Quality of Service Facilitator, hex 'FFFFF9'.

Payload: The format of the RSRM Extended Link Service command is shown in table 78.

Table 78 – RSRM payload

Item	Size (Bytes)
hex 'xx 00 00 00'	4
Reserved	1
Native Port Identifier of the proposed System Resource Manager	3

Native Port Identifier of the proposed System Resource Manager: The 24 bit Fibre Channel Address Identifier of the Class 4 Stream Port to serve as System Resource Manager.

Reply Extended Link Service Sequence:

Service Reject (LS_RJT)
 Signifies the rejection of the RSRM command
Accept (ACC)
 Signifies acceptance of the RSRM request

– Accept Payload

The format of the Accept Payload is shown in table 79.

Table 79 – RSRM Accept Payload

Item	Size (Bytes)
hex '02 00 00 00'	4

6.2.9 Suspend Stream

Class 4 Stream Definers and the System Resource Manager sends the Suspend Stream (C4_Pause) Extended Link Service command to deactivate, suspend, a Class 4 Stream, immediately or at a future date.

The C4_Pause command shall travel on Permanent Stream #2.

Protocol:

Single frame Suspend Stream request Sequence

Single frame Accept reply Sequence

Format: FT_1

Addressing: The S_ID shall designate a Class 4 Stream Definer or the System Resource Manager. The D_ID shall designate the System Resource Manager, if the C4_Pause command is sent by a Class 4 Stream Definer. If the C4_Pause command is sent by the System Resource Manager, then the D_ID shall designate the Class 4 Stream Producer.

Payload: The format of the C4_Pause Extended Link Service command is shown in table 80.

Table 80 – C4_Pause Payload

Item	Size (Bytes)
hex 'xx 00 00 00'	4
Reserved	1
Class 4 Stream Producer ID	3
Reserved	2
Tag	2
Suspension Time	4
Suspension Time Fraction	4

Class 4 Stream Producer ID: The source of the Class 4 Stream to be deactivated, suspended.

Tag: An unsigned 16 bit field identifying the Class 4 Stream to be suspended.

The Tag field shall be set to the value of the Stream Tag, if the C4_Pause command is sent by the Class 4 Stream Definer.

The Tag field shall be set to the value of the Producer Tag, if the C4_Pause command is sent by the System Resource Manager.

Suspension Time: This unsigned 32 bit field shall identify the time in seconds, relative to 00:00:00 Universal Time (UT) on 1 January 1900, at which transmission for the Class 4 Stream is to be suspended, deactivated.

Suspension Time Fraction: This unsigned 32 bit field shall identify an offset, in units of 1/4 294 967 297 second, to the Activation Time.

Immediate suspension of the Class 4 Stream is requested, if both the ‘Suspension Time’ and the ‘Suspension Time Fraction’ fields are set = hex ‘FF FF FF FF’.

The Suspension Time shall not be set so that both the ‘Suspension Time’ and the ‘Suspension Time Fraction’ fields are = hex ‘00 00 00 00’.

Reply Extended Link Service Sequence:

Service Reject (LS_RJT)
Signifies the rejection of the C4_Pause command

– Reject Reason code

The following Reject Reason codes, in addition to the FC–PH reason codes, are defined:

Invalid Class 4 Stream Producer ID: No Class 4 Stream Port with the given Fibre Channel address exist.

Invalid Tag: The Tag is not recognized by the C4_Pause command recipient, i.e. the recipient is not aware of any Class 4 Stream for the identified Class 4 Stream Producer using the given Tag value.

Invalid Pause Time: The time value specified by the ‘Suspension Time’ and ‘Suspension Time Fraction’ fields is illegal.

Request denied: The C4_Pause request have been denied for reason of security or management. However, the Class 4 Stream Producer shall not use this reason to reject a C4_Pause command sent by the System Resource Manager.

The Class 4 Stream Producer shall use this reason to reject any C4_Pause command not sent by the System Resource Manager.

Accept (ACC)
Signifies acceptance of the C4_Pause command

– Accept Payload

The format of the Accept Payload is shown in table 81.

Table 81 – C4_Pause Accept Payload

Item	Size (Bytes)
hex ‘02 00 00 00’	4

6.2.10 Tag Query

Class 4 Stream Definers sends the Tag Query (Tag_Query) Extended Link Service command to obtain the Class 4 Stream characteristics for a specific Stream Tag or Producer Tag associated with a given Class 4 Stream Producer.

The System Resource Manager sends the Tag Query Extended Link Service command to obtain the Class 4 Stream characteristics for a specific Producer Tag on a given Class 4 Stream Producer

The Tag_Query command shall travel on Permanent Stream #2.

Protocol:

Tag Query request Sequence
Accept reply Sequence

Format: FT_1

Addressing: The S_ID shall designate a Class 4 Stream Definer or the System Resource Manager. The D_ID shall designate the System Resource Manager, if the Tag_Query command is sent by a Class 4 Stream Definer. If the Tag_Query command is sent by the System Resource Manager, then the D_ID shall designate the Class 4 Stream Producer.

Payload: The format of the Tag_Query Extended Link Service command is shown in table 82.

Class 4 Stream Producer ID: The 24 bit Fibre Channel address for the Class 4 Stream Producer, for which the characteristics of a specific Class 4 Stream are sought.

Tag Type: The unsigned 8 bit identifier shall identify the type of the ‘Tag’ field, two values are defined for this field:

- *1:* Stream Tag;
- *2:* Producer Tag;
- *other values:* reserved.

Tag: The unsigned 16 bit identifier for the Class 4 Stream, whose characteristics are

Table 82 – Tag_Query Payload

Item	Size (Bytes)
hex ‘xx 00 00 00’	4
Reserved	1
Class 4 Stream Producer ID	3
Tag Type	1
Reserved	1
Tag	2

sought. This shall be a Stream Tag or a Producer Tag for a Tag_Query commands sent by a Class 4 Stream Definer and a Producer Tag for a Tag_Query commands sent by the System Resource Manager.

Reply Extended Link Service Sequence:

Service Reject (LS_RJT)

Signifies the rejection of the Tag_Query command

– Reject Reason code

The following Reject Reason code, in addition to the FC–PH reason codes, is defined:

Invalid Tag: The Tag is not recognized for the given Class 4 Stream Producer.

Accept (ACC)

Signifies acceptance of the Tag_Query command

– Accept Payload

The format of the Accept Payload is shown in table 83.

Start Stream Time: This unsigned 32 bit field shall identify the time in seconds relative to 00:00:00 Universal Time (UT) on 1 January 1900, at which transmission for the Class 4 stream is to begin.

Start Stream Time Fraction: This unsigned 32 bit field shall identify an offset, in units of 1/4 294 967 297 second, to the Start Stream Time.

No Start Stream Time are specified for the Class 4 Stream, if both the ‘Start Stream Time’ and the ‘Start Stream Time Fraction’ fields are set = hex ‘00 00 00 00’

Restart Stream Time #n: This unsigned 32 bit field shall identify the time in seconds relative to 00:00:00 Universal Time (UT) on 1 January 1900, at which transmission for the Class 4 Stream is to continue.

Restart Stream Time Fraction #n: This unsigned 32 bit field shall identify an offset, in units of 1/4 294 967 297 second, to the Restart Stream Time.

No Restart Stream Time are specified for the Class 4 Stream, if both the ‘Restart Stream Time #n’ and the ‘Restart Stream Time Fraction #n’ fields are set = hex ‘00 00 00 00’

Pause Stream Time #n: This unsigned 32 bit field shall identify the time in seconds relative to 00:00:00 Universal Time (UT) on 1 January 1900, at which transmission for the Class 4 Stream is to cease.

Pause Stream Time Fraction #n: This unsigned 32 bit field shall identify an offset, in units of 1/4 294 967 297 second, to the Pause Stream Time.

No Pause Stream Time are specified for the Class 4 Stream, if both the ‘Pause Stream Time #n’ and the ‘Pause Stream Time Fraction #n’ fields are set = hex ‘00 00 00 00’

Stop Stream Time: This unsigned 32 bit field shall identify the time in seconds relative to 00:00:00 Universal Time (UT) on 1 January 1900, at which transmission for the Class 4 Stream is to be terminated.

Stop Stream Time Fraction: This unsigned 32 bit field shall identify an offset, in units of 1/4 294 967 297 second, to the Stop Stream Time.

No Stop Time are specified for the Class 4 Stream, if both the ‘Stop Stream Time’ and the ‘Stop Stream Time Fraction’ fields are set = hex ‘00 00 00 00’

Stream Type: This unsigned 8 bit field shall identify Class 4 Stream Type to be used for this Class 4 Stream.

Three values are defined for the Stream Type variable; 1 for Class 4 Asynchronous, 2 for Class 4 Available Bit Rate and 3 for Class 4 Re-

Table 83 – Tag_Query Accept Payload

Item	Size (Bytes)
hex ‘02 00 00 00’	4
Start Stream Time	4
Start Stream Time Fraction	4
Restart Stream Time #2	4
Restart Stream Time Fraction #2	4
Restart Stream Time #3	4
Restart Stream Time Fraction #3	4
Restart Stream Time #4	4
Restart Stream Time Fraction #4	4
Restart Stream Time #5	4
Restart Stream Time Fraction #5	4
Restart Stream Time #6	4
Restart Stream Time Fraction #6	4
Restart Stream Time #7	4
Restart Stream Time Fraction #7	4
Restart Stream Time #8	4
Restart Stream Time Fraction #8	4
Pause Stream Time #1	4
Pause Stream Time Fraction #1	4
Pause Stream Time #2	4
Pause Stream Time Fraction #2	4
Pause Stream Time #3	4
Pause Stream Time Fraction #3	4
Pause Stream Time #4	4
Pause Stream Time Fraction #4	4
Pause Stream Time #5	4
Pause Stream Time Fraction #5	4
Pause Stream Time #6	4
Pause Stream Time Fraction #6	4
Pause Stream Time #7	4
Pause Stream Time Fraction #7	4
Pause Stream Time #8	4
Pause Stream Time Fraction #8	4
Stop Stream Time	4
Stop Stream Time Fraction	4
Stream Type	1
Reserved	2
Number of Component Streams	1
Component Stream parameter set #1	k
…	
Component Stream parameter set #n	l

al–Time Streams. All other values for the Stream Type field are reserved.

Number of assigned Tags: This unsigned 8 bit field shall identify the number of Component Stream used to carry the Class 4 Stream.

Component Stream parameter set: One or more sets of Component Stream parameters are defined, as specified by the ‘Class 4 Stream Consumer ID’ field. The format of the Component Stream parameter set is shown in table 84.

Maximum Bandwidth: This unsigned 32 bit field shall, in units of Bytes per second, identify the largest bandwidth allocated to the Component Stream.

Minimum Bandwidth: This unsigned 32 bit field shall, in units of Bytes per second, identify the smallest bandwidth allocated to the Component Stream.

Table 84 – Component Stream – Parameters

Item	Size (Bytes)
Maximum Bandwidth	4
Minimum Bandwidth	4
Maximum Data Field Size	2
Length of Component Stream ID	2
Component Stream Identifier	m

Stream Data Field Size: This unsigned 16 bit field shall represent the largest Payload size, as defined by FC–PH, to be used by this Component Stream.

Length of Component Stream ID: This unsigned 16 bit field shall identify the number of bytes in the 'Component Stream Identifier' field. This shall be a value equal to an integer number of words.

Component Stream Identifier: This field shall, in a vendor unique manner, identify the specific piece of Audio–Visual material or a specific Audio–Visual source within the Stream Producer.

6.2.11 Terminate Stream

Class 4 Stream Definers and the System Resource Manager sends the Terminate Stream (C4_Stop) Extended Link Service command to terminate a Class 4 Stream, immediately or at a future date.

The C4_Stop command shall travel on Permanent Stream #2.

Protocol:

Single frame Terminate Stream request Sequence
Single frame Accept reply Sequence

Format: FT_1

Addressing: The S_ID shall designate a Class 4 Stream Definer or the System Resource Manager. The D_ID shall designate the System Resource Manager, if the C4_Stop command is sent by a Class 4 Stream Definer. If the C4_Stop command is sent by the System Resource Manager, then the D_ID shall designate the Class 4 Stream Producer.

Payload: The format of the C4_Stop Extended Link Service command is shown in table 85.

Table 85 – C4_Stop Payload

Item	Size (Bytes)
hex 'xx 00 00 00'	4
Reserved	1
Class 4 Stream Producer ID	3
Reserved	2
Tag	2
Termination Time	4
Termination Time Fraction	4

Class 4 Stream Producer ID: This unsigned 24 bit field shall identify the source of the Class 4 Stream to be terminated.

Tag: An unsigned 16 bit field identifying the Class 4 Stream to be terminated.

The Tag field shall be set to the value of the Stream Tag, if the C4_Stop command is sent by the Class 4 Stream Definer.

The Tag field shall be set to the value of the Producer Tag, if the C4_Stop command is sent by the System Resource Manager.

Termination Time: This unsigned 32 bit field shall identify the time in seconds, relative to 00:00:00 Universal Time (UT) on 1 January 1900, at which transmission for the identified Class 4 Stream is to cease.

Termination Time Fraction: This unsigned 32 bit field shall identify an offset, in units of 1/4 294 967 297 second, to the Termination Time.

Immediate termination of the Class 4 Stream is requested, if both the 'Termination Time' and the 'Termination Time Fraction' fields are set = hex 'FF FF FF FF'.

The Termination Time shall not be set so that both the 'Termination Time' and the 'Termination Time Fraction' fields are = hex '00 00 00 00'.

Reply Extended Link Service Sequence:

Service Reject (LS_RJT)

Signifies the rejection of the C4_Stop command

– Reject Reason code

The following Reject Reason codes, in addition to the FC–PH reason codes, are defined:

Invalid Class 4 Stream Producer ID: No Class 4 Stream Port with the given Fibre Channel address exist.

Invalid Tag: The Tag is not recognized by the C4_Stop command recipient, i.e. the recipient is not aware of any Class 4 Stream for the identified Class 4 Stream Producer using the given Tag value.

Invalid Termination Time: The time value specified by the ‘Termination Time’ and ‘Termination Time Fraction’ fields is illegal.

Request denied: The C4_Stop request have been denied for reason of security or management. However, the Class 4 Stream Producer shall not use this reason to reject a C4_Stop command sent by the System Resource Manager.

The Class 4 Stream Producer shall use this reason to reject any C4_Stop command not sent by the System Resource Manager.

Accept (ACC)

Signifies acceptance of the C4_Stop command

– Accept Payload

The format of the Accept Payload is shown in table 86.

Table 86 – C4_Stop Accept Payload

Item	Size (Bytes)
hex ‘02 00 00 00’	4

7 Class 6 Multicast Streams

The following types of Streams are defined:

– *Asynchronous Stream: Used to carry intermittent, sporadic information between a Stream Producer and a Stream Consumer;*

– *Real–Time Stream: Used to carry Real–Time information, with guaranteed Quality of Service parameters, on one or more Exchanges between a Stream Producer and a Stream Consumer.*

7.1 Stream basics

Class 6 - Unidirectional Dedicated Connection service (see clause 22 of FC-PH-3), and Class 6 Multicast (see 31.2 of FC-PH-3) is a service that provides unidirectional, Dedicated Connections for reliable multicast through a Fabric.

The Class 6 Multicast streaming function requires an agreement among user nodes to abide by a set of streaming rules within a multicast group. The Fabric provides Class 6 services but does not provide enforcement of those rules required to maintain a streaming function. Other Classes of Service may coexist with Class 6 service on a single Fabric.

Four logical entities participate in the management and transfer of data streams. They are:

- Stream Consumer;
- Stream Definer;
- Stream Producer;
- System Resource Manager (SRM).

7.1.1 Stream Consumer

The Stream Consumer is that logical entity that receives a data Stream.

7.1.2 Stream Definer

The Stream Definer is that logical entity that initiates the establishment, suspension, resumption, and termination of data Streams.

7.1.3 Stream Producer

The Stream Producer is that logical entity that sources a data Stream. In Class 6 operation, the Stream Producer originates and tears down the Class 6 multicast connection.

7.1.4 System Resource Manager

The System Resource Manager is the logical entity that reserves bandwidth for Stream Producers and Stream Consumers at the request of a Stream Definer. The System Resource Manager also originates the Prompt Stream Control Service (see 7.3.5) used to create time slots for data streaming.

In Class 6 operation, the System Resource Manager organizes the Class 6 multicast groups as required by requests coming from a Stream Definer.

7.2 Predefined Quality of Service Parameters

Editor’s Note: TBD

7.3 Extended Link Service - Stream support

The following Extended Link Service commands are defined for the support of Class 6 Streams:

– *Get Stream Ability (GSA):* Enable a port to obtain the Stream capabilities and parameters of another port;

– *Locate System Resource Manager (LSRM):* Enable an Nx_Port to find the address identifier of the System Resource Manager;

– *Prompt:* Enable the System Resource Manger to initiate Stream flows with a single command;

– *Register System Resource Manager (RSRM):* Identify an Nx_Port as a System Resource Manager;

– *Stream Control Services (SCS):* Enables a Stream Definer to establish, suspend, resume or terminate Streams, and the System Resource Manager to notify the Stream Producer of third party establishment, suspension, resumption or termination of Streams;

Support for the above listed Extended Link Service commands shall not require use of PLOGI prior to their use.

7.3.1 Get Stream Ability

Ports send a Get Stream Ability (GSA) Extended Link Service command to a port whose Stream parameters are desired.

The System Resource Manager shall, except as noted below, obtain the Stream parameters of both the Stream Producer and the Stream Consumer before accepting any Stream requests for the Stream Producer and the Stream Consumer.

Stream Definers should, except as noted below, obtain the Stream parameters of both the Stream Producer and the Stream Consumer before issuing any Stream requests for the Stream Producer and the Stream Consumer.

Stream Producers shall obtain the Stream Consumers Stream parameters before accepting establishment of any Stream between the Stream Producer and the Stream Consumer.

Transmission of the GSA and the corresponding Accept shall be sent as an Asynchronous Stream in an individual Asynchronous Stream and is not subject to the above stated restrictions.

Protocol:

Get Stream Ability request Sequence
Accept reply Sequence

Format: FT_1

Addressing: The S_ID field designates the source wishing to obtain the destination's Stream parameters. The D_ID designates the port whose Stream parameters are desired.

Payload: The format of the GSA Extended Link Service command is shown in table 87.

Table 87 – GSA payload

Item	Size (Bytes)
hex ‘xx 00 00 00’	4

Reply Extended Link Service Sequence:

Service Reject (LS_RJT)
  Signifies the rejection of the GSA command
Accept (ACC)
  Signifies acceptance of the GSA request

– Accept Payload

The format of the Accept Payload is shown in table 88.

Table 88 – GSA Accept Payload

Item	Size (Bytes)
hex ‘02 00 00 00’	4
SP max. Class 6 RT Streams	1
SC max. Class 6 RT Streams	1
Reserved	2
Max. latency for Class 6 initial RT ACK	2
Max. latency for Class 6 final RT ACK	2
SC max. Class 6 RT buffer size	4

SP max. Class 6 RT Streams: This 8 bit unsigned field, shall indicate the maximum number of Class 6 Real–Time Streams the port is able to support as a Stream Producer.

SC max. Class 6 RT Streams: This 8 bit unsigned field, shall indicate the maximum number of Class 6 Real–Time Streams the port is able to support as a Stream Consumer.

Max. latency for Class 6 initial RT ACK: This 16 bit unsigned field shall indicate, in units of 1/65 536 of a second, the worst case latency before the first ACK or P_RJT for a Class 6 Sequence received in a Real–Time Stream is transmitted.

Max. latency for Class 6 final RT ACK: This 16 bit unsigned field shall indicate, in units of 1/65 536 of a second, the worst case latency before the final ACK or P_RJT for a Class 6 Sequence received in a Real–Time Stream is transmitted.

SC max. Class 6 RT buffer size:

Editor's Note: TBD

7.3.2 System Resource Manager – Locate

Ports send a Locate System Resource Manager (LSRM) Extended Link Service command to the Quality of Service Facilitator (QoSF), located at the well-known address hex ‘FFFFF9’, to find the Address Identifier for the System Resource Manager.

Protocol:

Locate System Resource Manager request Sequence
Accept reply Sequence

Format: FT_1

Addressing: The S_ID field designates the source Port requesting the Address Identifier. The D_ID designates the Quality of Service Facilitator, hex ‘FFFFF9’.

The LSRM request shall be rejected (LS_RJT) with a reason code explanation of ‘No System Resource Manager registered’ if the Quality of Service Facilitator is unaware of the presence of a System Resource Manager.

Payload: The format of the LSRM Extended Link Service command is shown in table 89.

Table 89 – LSRM payload

Item	Size (Bytes)
hex ‘xx 00 00 00’	4

Reply Extended Link Service Sequence:

Service Reject (LS_RJT)
Signifies the rejection of the LSRM command
Accept (ACC)
Signifies acceptance of the LSRM request

– Accept Payload

The format of the Accept Payload is shown in table 90.

Table 90 – LSRM Accept Payload

Item	Size (Bytes)
hex ‘02 00 00 00’	4
Reserved	1
Native Port Identifier of the System Resource Manager	3

Native Port Identifier of the System Resource Manager: The Address Identifier used by the System Resource Manager, to accept Stream Control Service (SCS) requests and transmit Stream notifications.

7.3.3 System Resource Manager – Prompt

The System Resource Manager broadcasts a Prompt Extended Link Service command, to mark the beginning of a new Time Slot.

Protocol:

Prompt request Sequence (Single frame)

Format: FT_1

Addressing: The D_ID field designates the well–known Broadcast address hex ‘FFFFFF’. The S_ID field designates the System Resource Manager.

Payload: The format of the Prompt Extended Link Service command is shown in table 91.

Table 91 – Prompt payload

Item	Size (Bytes)
hex ‘xx 00 00 00’	4
Time Slot Number	4
Stream Time	8

Time Slot Number: An unsigned 32 bit field indicating the activated Time Slot Number.

Stream Time: A 64 bit time value, using the format specified by FC–PH for the Expiration timer, providing the System Stream Time, for the use of Stream Definers and Stream Producers.

Reply Extended Link Service Sequence:

None

7.3.4 System Resource Manager – Register

Ports sends an Register System Resource Manager (RSRM) Extended Link Service command to the Quality of Service Facilitator (QoSF), located at the well-known address hex ‘FFFFF9’, to register their ability to become the System Resource Manager.

Protocol:

Register System Resource Manager request Sequence
Accept reply Sequence

Format: FT_1

Addressing: The S_ID field designates the source Port wishing to become the System Resource Manager. The D_ID designates the Quality of Service Facilitator, hex ‘FFFFF9’.

Payload: The format of the RSRM Extended Link Service command is shown in table 92.

Table 92 – RSRM payload

Item	Size (Bytes)
hex ‘xx 00 00 00’	4
Reserved	1
Native Port Identifier of the proposed System Resource Manager	3

Native Port Identifier of the proposed System Resource Manager: The Address Identifier to be used by the System Resource Manager, to accept Stream Control Service (SCS) requests and transmit Stream notifications.

Reply Extended Link Service Sequence:

Service Reject (LS_RJT)
Signifies the rejection of the RSRM command
Accept (ACC)
Signifies acceptance of the RSRM request

– Accept Payload

The format of the Accept Payload is shown in table 93.

Table 93 – RSRM Accept Payload

Item	Size (Bytes)
hex ‘02 00 00 00’	4

7.3.5 Stream Control Services

The Stream Control Services (SCS) Extended Link Service command shall be used by Stream Definers to request establishment, suspension, resumption and termination of Class 6 Streams. The SCS Extended Link Service command shall be used by the System Resource Manager to notify Stream Producers of third party establishment, suspension, resumption and termination request of Class 6 Streams.

Stream Definers may send a single SCS reservation request Sequence every Slot period. Stream Definers shall not send more than one SCS Extended Link Service Command per Time Slot period.

A SCS Sequence may contain multiple Stream requests. The SCS Sequence is limited in size to a maximum of 14 Stream requests. If multiple Stream requests are made in an SCS request Sequence, then they shall be sorted according to Priority order, with higher priority Stream requests appearing before lower order priority Stream requests. Furthermore, a Stream Definer is limited to a maximum of 256 pending Stream requests at any given time.

Protocol:

Register System Resource Manager request Sequence (Single frame)
Accept reply Sequence

Format: FT_1

Addressing: The S_ID field designates the Stream Definer, for Stream requests, or the System Resource Manager, for Stream notifications. The D_ID designates the System Resource Manager, if the S_ID identify a Stream Definer or a Stream Producer, if the S_ID identify the System Resource Manager.

Payload: The format of the SCS Extended Link Service command is shown in table 94.

Table 94 – SCS Payload

Item	Size (Bytes)
hex ‘xx 00 00 00’	4
Reserved	4
SCS request or notification #1	16
…	
SCS request or notification #n	16

SCS request or notification: If the SCS Extended Link Service Sequence is sent by a Stream Definer, then this field shall follow one of the Stream requests defined in 7.3.5.1. If the

SCS Extended Link Service Sequence is sent by a System Resource Manager, then this field shall follow one of the Stream notifications defined in 7.3.5.2.

Reply Extended Link Service Sequence:

Service Reject (LS_RJT)
Signifies the rejection of the SCS command
Accept (ACC)
Signifies acceptance of the SCS request

– Accept Payload

The format of the Accept Payload is shown in table 95.

Table 95 – SCS Accept Payload

Item	Size (Bytes)
hex ‘02 00 00 00’	4
Reserved	4
SCS response #1	16
...	
SCS response #m	16

SCS response: One of the SCS responses, defined in see 7.3.5.3.

NOTE – The number and order of the SCS responses does not match the SCS request number or order, as replies to SCS requests may appear in separate SCS Accept Sequences.

7.3.5.1 Stream Control Service – Requests

The following Stream reservation requests are defined:

– *No Operation (NOP):* Used to pass an empty Stream request to the System Resource Manager;

– *Preemptive Reserve Real–Time Stream (PRRTS):* Used to establish a Real–Time Stream between a Stream Producer and a Stream Consumer, preempting any lower priority Streams;

– *Reserve Real–Time Stream (RRTS):* Used to establish a Real–Time Stream between a Stream Producer and a Stream Consumer;

– *Resume Real–Time Stream (RSRTS):* Used to restart a suspended Real–Time Stream.

– *Suspend Real–Time Stream (SRTS):* Used to quiesce a Real–Time Stream;

– *Terminate Real–Time Stream (TRTS):* Used to remove an Real–Time Stream, releasing the resource allocated to the Stream.

7.3.5.1.1 No Operation

The No Operation (NOP) request is used by a Steam Definer, to pass an empty Stream request to the System Resource Manager.

Payload: The format of the NOP request is shown in table 96.

Table 96 – NOP (00)

Item	Size (Bytes)
Control	1
Stream Command Code (hex ‘00’)	1
Reserved	1
Transaction Identifier	1
Reserved	12

Control: An unsigned 8 bit field, consisting of two 4 bit sub-fields, a Priority field and a Class of Service field. The 4 bit Priority field is limited to the range of 0, highest priority, to 7, the lowest priority level. The Priority field uses bits 28 through 31 of the first word in the SCS request. The second field, the Class of Service field, using bits 24 through 27 in the first word, is reserved, and shall be set = 0.

Stream Command Code: An 8 bit field uniquely identifying the Stream request. The SCC for the NOP Stream request shall be hex ‘00’.

Transaction Identifier: An 8 bit identifier, assigned by the Stream Definer. The Transaction Identifier ensures that the Stream Definer can match the Accept or Reject Stream reply to the initial NOP Stream request.

Replies to Stream requests:

Stream request rejection
Signifies the rejection of the NOP request

– Reject payload; the format of the Reject response is shown in 7.3.5.3.2.

Stream request acceptance
Signifies acceptance of the NOP request

– Accept payload; the format of the Accept response is shown in subsequent clauses.

7.3.5.1.2 Preemptive Reserve Real–Time Stream

The Preemptive Reserve Real–Time Stream (PRRTS) Stream request is used by a Stream Definer to request reservation for the transfer of one or more Sequences, in a periodic Real–Time Time Slot pattern, from a Stream Producer, to a given destination, the Stream Consumer, starting at a specific time.

The System Resource Manager will, to successfully honor the Stream request, terminate lower priority Available Bit Rate or Real–Time Streams, blocking the PRRTS.

The Stream Definer, need not be the Stream Producer nor the Stream Consumer.

Payload: The format of the PRRTS Stream request is shown in table 97.

Table 97 – PRRTS (07)

Item	Size (Bytes)
Control	1
Stream Command Code (hex ‘07’)	1
Quality of Service Parameters	1
Transaction Identifier	1
Producer Tag	1
Stream Producer	3
Start Time fraction (1/256 s)	1
Stream Consumer	3
Start Time (s)	4

Control: An unsigned 8 bit field, consisting of two 4 bit sub-fields, a Priority field and a Class of Service field. The 4 bit Priority field is limited to the range of 0, highest priority, to 7, the lowest priority level. The Priority field uses bits 28 through 31 of the first word in the SCS request. The second field, using bits 24 through 27 in the first word, indicate the Class of Service which will be used to transmit the requested Stream. The encoding of the Class of Service field is as shown in table 98.

Table 98 – Class of Service – Encoding

Item	Code
Reserved	0 – 1
Class 2 Service	2
Class 3 Service	3
Reserved	4 – F

Stream Command Code: An 8 bit field uniquely identifying the Stream request. The SCC for the PRRTS Stream request shall be hex ‘07’.

Quality of Service Parameters: The 8 bit Quality of Service Parameter field define either a specific predefined set of Quality of Service Parameter (see 7.2) or form a pair of sub-fields specifying Jitter and Bandwidth requirements. The most significant bit, bit 15, in the Quality of Service Parameter field is used to determine, whether predefined Quality of Service parameters or separate Jitter and Bandwidth is being specified. If bit 15 is set = 1, then predefined Quality of Service, as defined in 7.2, is being requested. If bit 15 is set = 0 then Jitter and Bandwidth requirements are provided.

The unsigned 2 bit Jitter field, using bits 13 through 14, in the first word of the Stream request, define the maximum amount of source jitter, the System Resource Manager may introduce by means of Time Slot scheduling. Table 99 defines the maximum amount of Time Slot scheduling jitter the System Resource Manager may introduce for different values of the Jitter field.

The unsigned 5 bit Bandwidth field (N), using bits 8 through 12, in the first word of the Stream request, define the bandwidth requirement for the Real–Time Stream, in units of Time Slot per

Table 99 – SRM Jitter limit – Encoding

Jitter limit	Value
Source Jitter ≤ 1 Time Slot	0
Source Jitter ≤ 2 Time Slot	1
Source Jitter ≤ 4 Time Slot	2
Source Jitter ≤ 6 Time Slot	3

second.

$$N = \begin{cases} \text{Ceiling}\left(\dfrac{\text{Bandwidth (B/s)}}{1\,350\,000}\right) & \text{@ 1,062 5 GBaud} \\ \text{Ceiling}\left(\dfrac{\text{Bandwidth (B/s)}}{1\,151\,000}\right) & \text{@ 2,125 GBaud} \\ \text{Ceiling}\left(\dfrac{\text{Bandwidth (B/s)}}{753\,000}\right) & \text{@ 4,25 GBaud} \end{cases}$$

$$31 \geq N \geq 1$$

Transaction Identifier: An 8 bit identifier, assigned by the Stream Definer. The Transaction Identifier ensure that the Stream Definer can match the Accept or Reject Stream reply to the initial PRRTS Stream request.

Producer Tag: An unique 8 bit identifier, assigned by the Stream Producer to a specific Audio Visual piece of material or source, selected by a method outside the scope of this document.

Stream Producer: A 24 bit address identifier, indicating the source of the Real–Time Stream.

Start Time fraction: The Start Time Fraction field shall, in units of 1/256 s, indicate an offset, delay, from the Start Time field before Stream transfer is to commence.

Stream Consumer: A 24 bit address identifier, indicating the destination of the Real–Time Stream.

Start Time: The Start Time field indicates in System Stream Time seconds, the instant at which Stream transfer is to commence.

Replies to Stream requests:

Stream request rejection

Signifies the rejection of the PRRTS request

– Reject payload; the format of the Reject response is shown in 7.3.5.3.2.

Stream request acceptance

Signifies acceptance of the PRRTS request

– Accept payload; the format of the Accept response is shown in 7.3.5.3.1.2.

7.3.5.1.3 Reserve Real–Time Stream

The Reserve Real–Time Stream (RRTS) Stream request is used by a Stream Definer to request reservation for the transfer of one or more Sequences, in a periodic Real–Time Time Slot pattern, from a Stream Producer, to a given destination, the Stream Consumer, starting at a specific time.

The Stream Definer, need not be the Stream Producer nor the Stream Consumer.

Payload: The format of the RRTS Stream request is shown in table 100.

Table 100 – RRTS (03)

Item	Size (Bytes)
Control	1
Stream Command Code (hex ‘03’)	1
Quality of Service Parameters	1
Transaction Identifier	1
Producer Tag	1
Stream Producer	3
Start Time fraction (1/256 s)	1
Stream Consumer	3
Start Time (s)	4

Control: An unsigned 8 bit field, consisting of two 4 bit sub-fields, a Priority field and a Class of Service field. The 4 bit Priority field is limited to the range of 0, highest priority, to 7, the lowest priority level. The Priority field uses bits 28 through 31 of the first word in the SCS request. The second field, using bits 24 through 27 in the first word, indicate the Class of Service which will be used to transmit the requested

Stream. The encoding of the Class of Service field is as shown in table 98.

Stream Command Code: An 8 bit field uniquely identifying the Stream request. The SCC for the RRTS Stream request shall be hex ‘03’.

Quality of Service Parameters: The 8 bit Quality of Service Parameter field define either a specific predefined set of Quality of Service Parameter (see 7.2) or form a pair of sub-fields specifying Jitter and Bandwidth requirements. The most significant bit, bit 15, in the Quality of Service Parameter field is used to determine, whether predefined Quality of Service parameters or separate Jitter and Bandwidth is being specified. If bit 15 is set = 1, then predefined Quality of Service, as defined in 7.2, is being requested. If bit 15 is set = 0 then Jitter and Bandwidth requirements are provided.

The unsigned 2 bit Jitter field, using bits 13 through 14, in the first word of the Stream request, define the maximum amount of source jitter, the System Resource Manager may introduce by means of Time Slot scheduling. Table 99 defines the maximum amount of Time Slot scheduling jitter the System Resource Manager may introduce for different values of the Jitter field.

$$N = \begin{cases} \text{Ceiling}\left(\dfrac{\text{Bandwidth (B/s)}}{1\,350\,000}\right) & \text{@ 1,062 5 GBaud} \\ \text{Ceiling}\left(\dfrac{\text{Bandwidth (B/s)}}{1\,151\,000}\right) & \text{@ 2,125 GBaud} \\ \text{Ceiling}\left(\dfrac{\text{Bandwidth (B/s)}}{753\,000}\right) & \text{@ 4,25 GBaud} \end{cases}$$

$$31 \geq N \geq 1$$

The unsigned 5 bit Bandwidth field (N), using bits 8 through 12, in the first word of the Stream request, define the bandwidth requirement for the Real–Time Stream, in units of Time Slot per second.

Transaction Identifier: An 8 bit identifier, assigned by the Stream Definer. The Transaction Identifier ensure that the Stream Definer can match the Accept or Reject Stream reply to the initial RRTS Stream request.

Producer Tag: An unique 8 bit identifier, assigned by the Stream Producer to a specific Audio Visual piece of material or source, selected by a method outside the scope of this document.

Stream Producer: A 24 bit address identifier, indicating the source of the Real–Time Stream.

Start Time fraction: The Start Time Fraction field shall, in units of 1/256 s, indicate an offset, delay, from the Start Time field before Stream transfer is to commence.

Stream Consumer: A 24 bit address identifier, indicating the destination of the Real–Time Stream.

Start Time: The Start Time field indicates in System Stream Time seconds, the instant at which Stream transfer is to commence.

Replies to Stream requests:

Stream request rejection
Signifies the rejection of the RRTS request

– Reject payload; the format of the Reject response is shown in 7.3.5.3.2.

Stream request acceptance
Signifies acceptance of the RRTS request

– Accept payload; the format of the Accept response is shown in 7.3.5.3.1.4.

7.3.5.1.4 Resume Real–Time Stream

The Resume Real–Time Stream (RSRTS) Stream request is used by a Stream Definer to request resumption of a Real–Time Stream, from a given Stream Producer, at a specific time.

The Stream Definer, need not be the Stream Producer nor the Stream Consumer.

Payload: The format of the RSRTS Stream request is shown in table 101.

Stream Identifier: The unsigned 8 bit identifier, assigned by the System Resource Manager to the Real–Time Stream.

Stream Command Code: An 8 bit field uniquely identifying the Stream request. The SCC for the RSRTS Stream request shall be hex ‘13’.

Transaction Identifier: An 8 bit identifier, assigned by the Stream Definer. The Transaction Identifier ensure that the Stream Definer can match the Accept or Reject Stream reply to the initial RSRTS Stream request.

Table 101 – RSRTS (13)

Item	Size (Bytes)
Stream Identifier	1
Stream Command Code (hex ‘13’)	1
Reserved	1
Transaction Identifier	1
Reserved	1
Stream Producer	3
Resume Time fraction (1/256 s)	1
Reserved	3
Resume Time (s)	4

Stream Producer: A 24 bit address identifier, indicating the source of the Real–Time Stream.

Resume Time fraction: The Resume Time Fraction field shall, in units of 1/256 s, indicate an offset, delay, from the Resume Time field before Stream transfer is to continue.

Resume Time: The Resume Time field indicates in System Stream Time seconds, the instant at which Stream transfer is to continue.

Replies to Stream requests:

Stream request rejection

Signifies the rejection of the RSRTS request

– Reject payload; the format of the Reject response is shown in 7.3.5.3.2.

Stream request acceptance

Signifies acceptance of the RSRTS request

– Accept payload; the format of the Accept response is shown in 7.3.5.3.1.5.

7.3.5.1.5 Suspend Real–Time Stream

The Suspend Real–Time Stream (SRTS) Stream request is used by a Stream Definer to request quiescence of a Real–Time Stream, from a given Stream Producer, at a specific time. The Stream Definer, need not be the Stream Producer nor the Stream Consumer.

Payload: The format of the SRTS Stream request is shown in table 102.

Table 102 – SRTS (0B)

Item	Size (Bytes)
Stream Identifier	1
Stream Command Code (hex ‘0B’)	1
Reserved	1
Transaction Identifier	1
Reserved	1
Stream Producer	3
Suspend Time fraction (1/256 s)	1
Reserved	3
Suspend time (s)	4

Stream Identifier: The unsigned 8 bit identifier, assigned by the System Resource Manager to the Real–Time Stream.

Stream Command Code: An 8 bit field uniquely identifying the Stream request. The SCC for the SRTS Stream request shall be hex ‘0B’.

Transaction Identifier: An 8 bit identifier, assigned by the Stream Definer. The Transaction Identifier ensure that the Stream Definer can match the Accept or Reject Stream reply to the initial SRTS Stream request.

Stream Producer: A 24 bit address identifier, indicating the source of the Real–Time Stream.

Suspend Time fraction: The Suspension Time Fraction field shall, in units of 1/256 s, indicate an offset, delay, from the Suspension Time field before Stream transfer is to be suspended.

Suspend Time: The Suspension Time field indicates in System Stream Time seconds, the instant at which Stream transfer is to be suspended.

Replies to Stream requests:

Stream request rejection
Signifies the rejection of the SRTS request

– Reject payload; the format of the Reject response is shown in 7.3.5.3.2.

Stream request acceptance
Signifies acceptance of the SRTS request

– Accept payload; the format of the Accept response is shown in 7.3.5.3.1.7.

7.3.5.1.6 Terminate Real–Time Stream

The Terminate Real–Time Stream (TRTS) Stream request is used by a Stream Definer to request cessation of a Real–Time Stream, from a given Stream Producer, at a specific time.

The Stream Definer, need not be the Stream Producer nor the Stream Consumer.

Payload: The format of the TRTS Stream request is shown in table 103.

Table 103 – TRTS (0F)

Item	Size (Bytes)
Stream Identifier	1
Stream Command Code (hex ‘0F’)	1
Reserved	1
Transaction Identifier	1
Reserved	1
Stream Producer	3
Termination Time fraction (1/256 s)	1
Reserved	3
Termination Time (s)	4

Stream Identifier: The unsigned 8 bit identifier, assigned by the System Resource Manager to the Real–Time Stream.

Stream Command Code: An 8 bit field uniquely identifying the Stream request. The SCC for the TRTS Stream request shall be hex ‘0F’.

Transaction Identifier: An 8 bit identifier, assigned by the Stream Definer. The Transaction Identifier ensure that the Stream Definer can match the Accept or Reject Stream reply to the initial TRTS Stream request.

Stream Producer: A 24 bit address identifier, indicating the source of the Real–Time Stream.

Termination Time fraction: The Termination Time Fraction field shall, in units of 1/256 s, indicate an offset, delay, from the Termination Time field before Stream transfer is to cease.

Termination Time: The Termination Time field indicates in System Stream Time seconds, the instant at which Stream transfer is to cease.

Replies to Stream requests:

Stream request rejection
Signifies the rejection of the TRTS request

– Reject payload; the format of the Reject response is shown in 7.3.5.3.2.

Stream request acceptance
Signifies acceptance of the TRTS request

– Accept payload; the format of the Accept response is shown in 7.3.5.3.1.9.

7.3.5.2 Stream Control Service – Notifications

SCS notifications are used by the System Resource Manager to inform the Stream Producer of Stream request made by a Stream Definer, other than the Stream Producer.

The following SCS notification requests are defined:

– *Real–Time Stream Notification (RTSN):* Used by the System Resource Manager to prepare the Stream Producer to send a Real–Time Stream.

– *Resume Real–Time Stream Notification (RSRTSN):* Used by the System Resource Manager to prepare the Stream Producer to resume transmission of a Real–Time Stream.

– *Suspend Real–Time Stream Notification (SRTSN):* Used by the System Resource Manager to inform the Stream Producer to suspend transmission of a Real–Time Stream.

– *Terminate Real–Time Stream Notification (TRTSN):* Used by the System Resource Manager to inform the Stream Producer to stop transmission of a Real–Time Stream.

7.3.5.2.1 Real–Time Stream Notification

The Real–Time Stream Notification (RTSN), is sent by the System Resource Manager to the Stream Producer, unless it was the Stream Definer.

The RTSN Stream request shall serve to inform the Stream Producer of the creation of a Real–Time Stream at a specific future date.

The RTSN shall be sent at least 1/16 second before the Stream Producer is required to send Real–Time information for the first Time Slot.

Payload: The format of the RTSN Stream notification is shown in table 104.

Table 104 – RTSN (23)

Item	Size (Bytes)
Stream Identifier	1
Stream Command Code (hex ‘23’)	1
Quality of Service Parameters	1
Transaction Identifier	1
Producer Tag	1
Stream Producer	3
Reserved	1
Stream Consumer	3
Start Time Slot	4

Stream Identifier: The unsigned 8 bit identifier, assigned by the System Resource Manager to the Real–Time Stream. Used by the Stream Producer to locate the Stream Slot Counter.

Stream Command Code: An 8 bit field uniquely identifying the Stream request. The SCC for the RTSN request shall be hex ‘23’.

Transaction Identifier: An 8 bit identifier, assigned by the System Resource Manager. The Transaction Identifier ensure that the System Resource Manager can match the Accept or Reject Stream reply to the RTSN Stream request.

Producer Tag: An unique 8 bit identifier, assigned by the Stream Producer to a specific Audio Visual piece of material or source, selected by a method outside the scope of this document.

Stream Producer: A 24 bit address identifier, indicating the source of the Real–Time Stream.

Stream Consumer: A 24 bit address identifier, indicating the destination of the Real–Time Stream.

Start Time Slot: The Slot Time, for the Slot Time Counter identified by the Stream Identifier, at which Real–Time transmission is to commence.

Replies to Stream notifications:

Stream notification rejection
Signifies the rejection of the RTSN request

– Reject payload; the format of the Reject response is shown in 7.3.5.3.2.

Stream notification acceptance
Signifies acceptance of the RTSN request

– Accept payload; the format of the Accept response is shown in 7.3.5.3.1.3.

7.3.5.2.2 Resume Real–Time Stream Notification

The Resume Real–Time Stream Notification (RSRTSN), is sent by the System Resource Manager to the Stream Producer, unless it was the Stream Definer.

The RSTRSN Stream request shall serve to inform the Stream Producer of the continuation of a Real–Time Stream at a specific future date.

The RSRTSN shall be sent at least 1/16 second before the Stream Producer is required to send Real–Time information for the first Time Slot.

Payload: The format of the RSRTSN Stream notification is shown in table 105.

Stream Identifier: The unsigned 8 bit identifier, assigned by the System Resource Manager to the Real–Time.

Stream Command Code: An 8 bit field uniquely identifying the Stream request. The SCC for the RSRTSN request shall be hex ‘33’.

Transaction Identifier: An 8 bit identifier, assigned by the System Resource Manager. The Transaction Identifier ensure that the System Resource Manager can match the Accept or

Table 105 – RSRTSN (33)

Item	Size (Bytes)
Stream Identifier	1
Stream Command Code (hex ‘33’)	1
Reserved	1
Transaction Identifier	1
Reserved	1
Stream Producer	3
Reserved	4
Resume Time Slot	4

Reject Stream reply to the RSRTSN Stream request.

Stream Producer: A 24 bit address identifier, indicating the source of the Real–Time Stream.

Stream Consumer: A 24 bit address identifier, indicating the destination of the Real–Time Stream.

Resume Time Slot: The Slot Time, for the Slot Time Counter identified by the Stream Identifier, at which Real–Time transmission is to continue.

Replies to Stream notifications:

Stream notification rejection
  Signifies the rejection of the RSRTSN request

– Reject payload; the format of the Reject response is shown in 7.3.5.3.2.

Stream notification acceptance
  Signifies acceptance of the RSRTSN request

– Accept payload; the format of the Accept response is shown in 7.3.5.3.1.6.

7.3.5.2.3 Suspend Real–Time Stream Notification

The Suspend Real–Time Stream Notification (SRTSN), is sent by the System Resource Manager to the Stream Producer, unless it was the Stream Definer.

The SRTSN Stream request shall serve to inform the Stream Producer of the suspension of a Real–Time Stream at a specific future date.

The SRTSN shall be sent at least 1/16 second before the Stream Producer is required to cease sending Real–Time information for the last Time Slot.

Payload: The format of the SRTSN Stream notification is shown in table 106.

Table 106 – SRTSN (2B)

Item	Size (Bytes)
Stream Identifier	1
Stream Command Code (hex ‘2B’)	1
Reserved	1
Transaction Identifier	1
Reserved	1
Stream Producer	3
Reserved	4
Suspend Time Slot	4

Stream Identifier: The unsigned 8 bit identifier, assigned by the System Resource Manager to the Real–Time Stream.

Stream Command Code: An 8 bit field uniquely identifying the Stream request. The SCC for the SRTSN request shall be hex ‘2B’.

Transaction Identifier: An 8 bit identifier, assigned by the System Resource Manager. The Transaction Identifier ensure that the System Resource Manager can match the Accept of Reject Stream reply to the SRTSN Stream request.

Stream Producer: A 24 bit address identifier, indicating the source of the Real–Time Stream.

Stream Consumer: A 24 bit address identifier, indicating the destination of the Real–Time Stream.

Suspend Time Slot: The Slot Time, for the Slot Time Counter identified by the Stream Identifier, at which Real–Time transmission is to cease.

Replies to Stream notifications:

Stream notification rejection

Signifies the rejection of the SRTSN request

– Reject payload; the format of the Reject response is shown in 7.3.5.3.2.

Stream notification acceptance

Signifies acceptance of the SRTSN request

– Accept payload; the format of the Accept response is shown in 7.3.5.3.1.8.

7.3.5.2.4 Terminate Real–Time Stream Notification

The Terminate Real–Time Stream Notification (TRTSN), is sent by the System Resource Manager to the Stream Producer, unless it was the Stream Definer.

The TRTSN Stream request shall serve to inform the Stream Producer of the cessation of a Real–Time Stream at a specific future date, at which time the resources assigned to the Real–Time Stream are released.

The TRTSN shall be sent at least 1/16 second before the Stream Producer is required to send Real–Time information for the last Time Slot.

Payload: The format of the TRTSN Stream notification is shown in table 107.

Table 107 – TRTSN (2F)

Item	Size (Bytes)
Stream Identifier	1
Stream Command Code (hex ‘2F’)	1
Reserved	1
Transaction Identifier	1
Reserved	1
Stream Producer	3
Reserved	4
Termination Time Slot	4

Stream Identifier: The unsigned 8 bit identifier, assigned by the System Resource Manager to the Real–Time Stream.

Stream Command Code: An 8 bit field uniquely identifying the Stream request. The SCC for the TRTSN request shall be hex ‘2F’.

Transaction Identifier: An 8 bit identifier, assigned by the System Resource Manager. The Transaction Identifier ensure that the System Resource Manager can match the Accept or Reject Stream reply to the TRTSN Stream request.

Stream Producer: A 24 bit address identifier, indicating the source of the Real–Time Stream.

Terminate Time Slot: The Slot Time, for the Slot Time Counter identified by the Stream Identifier, at which Real–Time transmission is to finish.

Replies to Stream notifications:

Stream notification rejection

Signifies the rejection of the TRTSN request

– Reject payload; the format of the Reject response is shown in 7.3.5.3.2.

Stream notification acceptance

Signifies acceptance of the TRTSN request

– Accept payload; the format of the Accept response is shown in 7.3.5.3.1.10.

7.3.5.3 Stream Control Service – Replies

Two types of replies are defined; acceptance and rejection. The acceptance replies shall signify that the SCS Stream request or Stream notification for which it is returned is valid and have been accepted. The rejection reply shall signify that the SCS Stream request or Stream notification for which it is returned is invalid and have been rejected.

7.3.5.3.1 Stream Control Service – Acceptance replies

The following SCS Stream acceptance replies are defined:

– *Accept Available Bit Rate Stream Notification (AABRSN):* Used by s Stream Producer to signal its acceptance of an ABRSN Stream notification;

- *Accept No Operation (ANOP):* Used by the System Resource Manager to signal its acceptance of a NOP Stream request;

- *Accept Preemptive Reserve Real–Time Stream (APRRTS):* Used by the System Resource Manager to signal its acceptance of a PRRTS Stream request;

- *Accept Real–Time Stream Notification (ARTSN):* Used by a Stream Producer to signal its acceptance of a RTSN Stream notification;

- *Accept Reserve Real–Time Stream (ARRTS):* Used by the System Resource Manager to signal its acceptance of a RRTS Stream request;

- *Accept Resume Real–Time Stream (ARSRTS):* Used by the System Resource Manager to signal its acceptance of a RSRTS Stream request;

- *Accept Resume Real–Time Stream Notification (ARSRTSN):* Used be a Stream Producer to signal its acceptance of a RSRTSN Stream notification;

- *Accept Suspend Real–Time Stream (ASRTS):* Used by the System Resource Manager to signal its acceptance of a SRTS Stream request;

- *Accept Suspend Real–Time Stream Notification (ASRTSN):* Used by a Stream Producer to signal its acceptance of a SRTSN Stream notification;

- *Accept Terminate Real–Time Stream (ATRTS):* Used by the System Resource Manager to signal its acceptance of a TRTS Stream request;

- *Accept Terminate Real–Time Stream Notification (ATRTSN):* Used by a Stream Producer to signal its acceptance of a TRTSN Accept – No Operation

The Accept No Operation Stream (ANOP) acceptance reply shall signify that the System Resource Manager has accepted the NOP Stream request, for which it is being returned.

Payload: The format of the Accept to the NOP request is shown in table 108.

Table 108 – NOP Accept (40)

Item	Size (Bytes)
Reserved	1
Stream Command Code (hex ‘40’)	1
Reserved	1
Transaction Identifier	1
Reserved	12

Stream Command Code: An 8 bit field uniquely identifying the Stream acceptance reply. The SCC for the Accept to the NOP Stream request shall be hex ‘40’.

Transaction Identifier: The 8 bit identifier, assigned by the Stream Definer to the initial NOP Stream request.

7.3.5.3.1.1 Accept – Preemptive Reserve Available Bit Rate Stream

The Accept Preemptive Reserve Available Bit Rate Stream (APRABRS) acceptance reply shall signify that the System Resource Manager has accepted the PRABRS Stream request, for which it is being returned, having allocated the required resource to handle an Available Bit Rate Stream.

Payload: The format of the Accept to the PRABRS request is shown in table 109.

Table 109 – PRABRS Accept (46)

Item	Size (Bytes)
Stream Identifier	1
Stream Command Code (hex ‘46’)	1
Reserved	1
Transaction Identifier	1
Reserved	8
Start Time Slot	4

Stream Identifier: The unsigned 8 bit identifier, assigned by the System Resource Manager to the Available Bit Rate Stream. Used by the Stream Producer to locate the Stream Slot Counter.

Stream Command Code: An 8 bit field uniquely identifying the Stream acceptance reply. The SCC for the Accept to the PRABRS Stream request shall be hex ‘46’.

Transaction Identifier: The 8 bit identifier, assigned by the Stream Definer to the initial PRABRS Stream request.

Start Time Slot: The Slot Time, for the Slot Time Counter identified by the Stream Identifier, at which Available Bit Rate transmission is to commence.

7.3.5.3.1.2 Accept – Preemptive Reserve Real–Time Stream

The Accept Preemptive Reserve Real–Time Stream (APRRTS) acceptance reply shall signify that the System Resource Manager has accepted the PRRTS Stream request, for which it is being returned, having allocated the required resource to handle the Real–Time Stream.

Payload: The format of the Accept to the PRRTS request is shown in table 110.

Table 110 – PRRTS Accept (47)

Item	Size (Bytes)
Stream Identifier	1
Stream Command Code (hex ‘47’)	1
Reserved	1
Transaction Identifier	1
Reserved	8
Start Time Slot	4

Stream Identifier: The unsigned 8 bit identifier, assigned by the System Resource Manager to the Real–Time Stream. Used by the Stream Producer to locate the Stream Slot Counter.

Stream Command Code: An 8 bit field uniquely identifying the Stream acceptance reply. The SCC for the Accept to the PRRTS Stream request shall be hex ‘47’.

Transaction Identifier: The 8 bit identifier, assigned by the Stream Definer to the initial PRRTS Stream request.

Start Time Slot: The Slot Time, for the Slot Time Counter identified by the Stream Identifier, at which Real–Time transmission is to commence.

7.3.5.3.1.3 Accept – Real–Time Stream Notification

The Accept Real–Time Stream Notification (ARTSN) acceptance reply shall signify that the Stream Producer, sending the ARTSN, has accepted the RTSN Stream notification, for which it is being returned, having allocated the required resource to handle the Real–Time Stream.

Payload: The format of the Accept to the RTSN request is shown in table 111.

Table 111 – RTSN Accept (63)

Item	Size (Bytes)
Reserved	1
Stream Command Code (hex ‘63’)	1
Reserved	1
Transaction Identifier	1
Reserved	12

Stream Command Code: An 8 bit field uniquely identifying the Stream acceptance reply. The SCC for the Accept to the RTSN Stream notification shall be hex ‘63’.

Transaction Identifier: The 8 bit identifier, assigned by the System Resource Manager to the initial RTSN Stream notification.

7.3.5.3.1.4 Accept – Reserve Real–Time Stream

The Accept Reserve Real–Time Stream (ARRTS) acceptance reply shall signify that the System Resource Manager has accepted the RRTS Stream request, for which it is being returned, having allocated the required resource to handle the Real–Time Stream.

Payload: The format of the Accept to the RRTS request is shown in table 112.

Table 112 – RRTS Accept (43)

Item	Size (Bytes)
Stream Identifier	1
Stream Command Code (hex ‘43’)	1
Reserved	1
Transaction Identifier	1
Reserved	8
Start Time Slot	4

Stream Identifier: The unsigned 8 bit identifier, assigned by the System Resource Manager to the Real–Time Stream. Used by the Stream Producer to locate the Stream Slot Counter.

Stream Command Code: An 8 bit field uniquely identifying the Stream acceptance reply. The SCC for the Accept to the RRTS Stream request shall be hex ‘43’.

Transaction Identifier: The 8 bit identifier, assigned by the Stream Definer to the initial RRTS Stream request.

Start Time Slot: The Slot Time, for the Slot Time Counter identified by the Stream Identifier, at which Real–Time transmission is to commence.

7.3.5.3.1.5 Accept – Resume Real–Time Stream

The Accept Resume Real–Time Stream (ARSRTS) acceptance reply shall signify that the System Resource Manager has accepted the RSRTS Stream request, for which it is being returned.

Payload: The format of the Accept to the RSRTS request is shown in table 113.

Stream Identifier: The unsigned 8 bit identifier, assigned by the System Resource Manager to the Real–Time Stream.

Stream Command Code: An 8 bit field uniquely identifying the Stream acceptance reply. The SCC for the Accept to the RSRTS Stream request shall be hex ‘53’.

Table 113 – RSRTS Accept (53)

Item	Size (Bytes)
Stream Identifier	1
Stream Command Code (hex ‘53’)	1
Reserved	1
Transaction Identifier	1
Reserved	8
Resume Time Slot	4

Transaction Identifier: The 8 bit identifier, assigned by the Stream Definer to the initial RSRTS Stream request.

Resume Time Slot: The Slot Time, for the Slot Time Counter identified by the Stream Identifier, at which Real–Time transmission is to continue.

7.3.5.3.1.6 Accept – Resume Real–Time Stream Notification

The Accept Resume Real–Time Stream Notification (ARSRTSN) acceptance reply shall signify that the Stream Producer, sending the ARSRTSN, has accepted the RSRTSN Stream notification, for which it is being returned.

Payload: The format of the Accept to the RSRTSN request is shown in table 114.

Table 114 – RSRTSN Accept (73)

Item	Size (Bytes)
Reserved	1
Stream Command Code (hex ‘73’)	1
Reserved	1
Transaction Identifier	1
Reserved	12

Stream Command Code: An 8 bit field uniquely identifying the Stream acceptance reply. The SCC for the Accept to the RSRTSN Stream notification shall be hex ‘73’.

Transaction Identifier: The 8 bit identifier, assigned by the System Resource Manager to the initial RSRTSN Stream notification.

7.3.5.3.1.7 Accept – Suspend Real–Time Stream

The Accept Suspend Real–Time Stream (ASRTS) acceptance reply shall signify that the System Resource Manager has accepted the SRTS Stream request, for which it is being returned.

Payload: The format of the Accept to the SRTS request is shown in table 115.

Table 115 – SRTS Accept (4B)

Item	Size (Bytes)
Stream Identifier	1
Stream Command Code (hex ‘4B’)	1
Reserved	1
Transaction Identifier	1
Reserved	8
Suspend Time Slot	4

Stream Identifier: The unsigned 8 bit identifier, assigned by the System Resource Manager to the Real–Time Stream.

Stream Command Code: An 8 bit field uniquely identifying the Stream acceptance reply. The SCC for the Accept to the SRTS Stream request shall be hex ‘4B’.

Transaction Identifier: The 8 bit identifier, assigned by the Stream Definer to the initial SRTS Stream request.

Suspend Time Slot: The Slot Time, for the Slot Time Counter identified by the Stream Identifier, at which Real–Time transmission is to cease.

7.3.5.3.1.8 Accept – Suspend Real–Time Stream Notification

The Accept Suspend Real–Time Stream Notification (ASRTSN) acceptance reply shall signify that the Stream Producer, sending the ASRTSN, has accepted the SRTSN Stream notification, for which it is being returned.

Payload: The format of the Accept to the SRTSN request is shown in table 116.

Table 116 – SRTSN Accept (6B)

Item	Size (Bytes)
Reserved	1
Stream Command Code (hex ‘6B’)	1
Reserved	1
Transaction Identifier	1
Reserved	12

Stream Command Code: An 8 bit field uniquely identifying the Stream acceptance reply. The SCC for the Accept to the SRTSN Stream notification shall be hex ‘6B’.

Transaction Identifier: The 8 bit identifier, assigned by the System Resource Manager to the initial SRTSN Stream notification.

7.3.5.3.1.9 Accept – Terminate Real–Time Stream

The Accept Terminate Real–Time Stream (ATRTS) acceptance reply shall signify that the System Resource Manager has accepted the TRTS Stream request, for which it is being returned.

Payload: The format of the Accept to the TRTS request is shown in table 117.

Table 117 – TRTS Accept (4F)

Item	Size (Bytes)
Stream Identifier	1
Stream Command Code (hex ‘4F’)	1
Reserved	1
Transaction Identifier	1
Reserved	8
Termination Time Slot	4

Stream Identifier: The unsigned 8 bit identifier, assigned by the System Resource Manager to the Real–Time Stream.

Stream Command Code: An 8 bit field uniquely identifying the Stream acceptance reply. The SCC for the Accept to the TRTS Stream request shall be hex ‘4F’.

Transaction Identifier: The 8 bit identifier, assigned by the Stream Definer to the initial TRTS Stream request.

Termination Time Slot: The Slot Time, for the Slot Time Counter identified by the Stream Identifier, at which Real–Time transmission is to finish, releasing the resources assigned to the Real–Time Stream.

7.3.5.3.1.10 Accept – Terminate Real–Time Stream Notification

The Accept Terminate Real–Time Stream Notification (ATRTSN) acceptance reply shall signify that the Stream Producer, sending the ATRTSN, has accepted the TRTSN Stream notification, for which it is being returned.

Payload: The format of the Accept to the TRTSN request is shown in table 118.

Table 118 – TRTSN Accept (6F)

Item	Size (Bytes)
Reserved	1
Stream Command Code (hex ‘6F’)	1
Reserved	1
Transaction Identifier	1
Reserved	12

Stream Command Code: An 8 bit field uniquely identifying the Stream acceptance reply. The SCC for the Accept to the TRTSN Stream notification shall be hex ‘6F’.

Transaction Identifier: The 8 bit identifier, assigned by the System Resource Manager to the initial TRTSN Stream notification.

7.3.5.3.2 Stream Control Service – Rejection reply

The Stream reject (S_RJT) rejection reply shall signify, that the Stream Producer, sending the S_RJT, has rejected the Stream notification, for which it is being returned. The S_RJT rejection reply shall signify, that the System Resource Manager, sending the S_RJT, has rejected the Stream request, for which it is being returned.

Payload: The format of the Reject to a Stream request or Stream notification is shown in table 119.

Table 119 – Stream Reject (80)

Item	Size (Bytes)
Reason Code	1
Stream Command Code (hex ‘80’)	1
Reserved	1
Transaction Identifier	1
Reserved	12

Reason Code: The unsigned 8 bit Reason Code field identify why a given Stream request or Stream notification could not get honored. The lager valued Reason Code shall be selected if multiple Reason Codes apply.

Table 120 shows the defined Reason Codes.

Stream Command Code: An 8 bit field uniquely identifying the Stream reply. The SCC for the S_RJT rejection reply shall be hex ‘80’.

Transaction Identifier: The 8 bit identifier, assigned by the Stream Definer or the System Resource Manager to the initial Stream request or Stream notification. The Transaction Identifier ensure that the Stream Definer or the System Resource Manager can match the Reject Stream reply to the initial Stream request or notification.

Table 120 – Reject Reason Codes

Encoded value (Hex)	Description
00	Reserved
01	Invalid Priority
02	Invalid Class of Service
03	Invalid Stream Identifier
04	Invalid Stream Command Code
05	Invalid Jitter
06	Invalid Bandwidth
07	Invalid Real–Time Stream Type
08	Invalid Transaction Identifier
09	Invalid Producer Tag
0A	Invalid Stream Producer
0B	Invalid Start Time fraction
0C	Invalid Termination Time fraction
0D	Invalid Suspend Time fraction
0E	Invalid Termination Time fraction
0F	Invalid Stream Consumer
10	Invalid Number of Sequences
11	Invalid Total Number of Data Frames
12	Invalid Start Time unit
13	Invalid Termination Time unit
14	Invalid Suspend Time unit
15	Invalid Termination Time unit
16	Invalid Start Time Slot
17	Invalid Termination Time Slot
18	Invalid Suspend Time Slot
19	Invalid Termination Time Slot
1A–1F	Reserved
20	Stream Producer unable to source another ABR or Real–Time Stream
21	Stream Producer unable to source requested bandwidth
22	Stream Consumer unable to accept another ABR or Real–Time Stream
23	Stream Consumer unable to accept requested bandwidth
24	Insufficient System bandwidth to honor request
25	Insufficient System Resource Manager resources to queue request
26–FF	Reserved

8 Application Interface (API)

This clause describe an API (Stream Driver), supporting management of Class 2 and 3 Stream operation. However, functions related to transfer of information, data on the Asynchronous, Available Bit Rate and Real–Time Streams, are absent from this clause.

Editor's Note: Volunteers are needed to propose an API for the transfer of data on Asynchronous, ABR and Real–Time Streams.

8.1 API – Data structures and types

The Stream Driver API uses a number of data types and structure. The following data types are defined:

- *SERR:* serves to carry the return value of the different Stream Driver functions (see 8.1.1);
- *SINT:* used to define most of the Stream Driver variables (see 8.1.2).

The following data structures are defined:

- *QUERY:* holds the parameters and results for the StreamQuery function (see 8.1.4);
- *STIME:* define time (see 8.1.3);
- *STREAM:* defines a Stream (see 8.1.5);
- *Stream_Capabilities:* defines the Stream capabilities of a specific Port (see 8.1.6);
- *Stream_Consumer:* defines the Stream capabilities of a Stream Consumer (see 8.1.7);
- *Stream_Producer:* defines the Stream capabilities of a Stream Producer (see 8.1.8).

8.1.1 Data type – SERR

The SERR data type is used to convey completion status for the defined Stream Driver functions (see 8.2).

SERR shall be able to carry an unsigned 16 bit integer value[1)]. An example of a type definition for the SERR data type is shown in figure 3.

```
typedef unsigned int SERR;
```

Figure 3 – SERR

The defined values for the SERR return variable are shown in table 121.

8.1.2 Data type – SINT

The SINT data type is used to define most of the variable used by the Stream Driver API.

SINT shall be able to carry an unsigned 32 bit integer value[2)] , an example of a type definition for the SINT data type is shown in figure 4.

8.1.3 Structure – STIME

The STIME structure is used to define time. The format of the STIME data structure is shown in figure 5.

Time: This variable shall define time in seconds, relative to 00:00:00 Universal Time (UT) on 1 January 1900.

Fraction: Is the fractional part of a second in units of 1/4 294 967 296 second.

8.1.4 Structure – QUERY

This structure define the characteristics of the individual Streams assigned to a specific Stream Identifier at a given point in time. The format of the QUERY data structure is shown in figure 6.

StreamID: This variable shall identify the Stream, whose affiliations are sought.

SampleTime: This STIME structure shall, in System Stream Time, indicate the instant at which the information in the Results array was gathered.

Length: This variable shall define, before a call to the StreamQuery function, the size of the Results array (QUERY_SIZE), and, after a call to the StreamQuery function, the number of entries in the Results array.

Results[]: Each entry in this array structure shall hold the characteristics of a single Stream assigned to the given Stream Identifier.

Results[x].StreamType: This variable shall, for entry 'x', identify the type of Stream estab-

---

[1)] Either 16, 32 or 64 bit may be employed.
[2)] Either 32 or 64 bit may be employed.

Table 121 – SERR – Values

Encoded value (Hex)	Description
0	No error (successful completion)
1–FF	See 5.3.6.3.2 and table 53
100	Unsuccessful FLOGI
101	System Resource Manager not found
102	Name Server registration failure
103	Invalid Port Identifier
104	Unable to get Stream Capabilities from the Port
105	Invalid Pause Time
106	Invalid Start Time
107	Invalid Preemption parameter
108	Invalid size of Query Results array
109	Overflow
10A	Invalid Audio Visual source specification
10B	Unable to assign Producer Tag
10C	Invalid Resume Time
10D	Invalid Stop Time
10E	Invalid Stream Type
10F	System Stream Time unavailable
8000–FFFF	Vendor unique
Others	Reserved

```
typedef unsigned int SINT;
```

Figure 4 – SINT

```
typedef struct {
  SINT Time;
  SINT Fraction;
} STIME;
```

Figure 5 – STIME structure

```
typedef struct {
  SINT StreamID;
  STIME SampleTime;
  SINT Length;
  struct {
       SINT StreamType;
       SINT StreamConsumerID;
       SINT StreamProducerID;
       SINT Priority;
       SINT ClassOfService;
       STIME StartTime;
       STIME StopTime;
       STIME PauseTime;
       STIME ResumeTime;
       } Results[QUERY_SIZE];
} QUERY;
```

Figure 6 – QUERY structure

lished between the Stream Producer and the Stream Consumer.

Two StreamType values are defined, Available Bit Rate Stream (ST_ABR) and Real–Time Stream (ST_RT).

Results[x].StreamConsumerID: This variable shall, for entry ‘x’, identify the destination of the specific Stream.

Results[x].StreamProducerID: This variable shall, for entry ‘x’, identify the source of the specific Stream.

Results[x].Priority: This variable shall, for entry ‘x’, identify the priority, used when establishing this Stream.

Eight priority levels are defined, in the range from 0, highest priority, to 7, lowest priority.

Results[x].ClassOfService: This variable shall identify the Class of Service used to carry data for the Stream between the Stream Producer and the Stream Consumer.

Two values are defined for this variable; 2 (Class 2) and 3 (Class 3).

Results[x].StartTime: This structure shall, for entry ‘x’, define the time, in System Stream

Time, at which transmission has begun or will begin for the specific Stream.

Results[x].StopTime: This structure shall, for entry ‘x’, define the time, in System Stream Time, at which transmission will be terminated for the specific Stream.

If no StopTime is specified, then the Time component of the StopTime structure is set = 0.

This structure shall on be meaningful for Available Bit Rate and Real–Time Streams.

Results[x].PauseTime: This structure shall, for entry ‘x’, define the time, in System Stream Time, at which transmission has been or will be suspended for the specific Stream.

This structure shall only be meaningful for Real–Time Streams.

If no PauseTime is specified, then the Time component of the PauseTime structure is set = 0.

Results[x].ResumeTime: This structure shall, for entry ‘x’, define the time, in System Stream Time, at which transmission has been or will be resumed for the specific Stream.

This structure shall only be meaningful for Real–Time Streams.

If no ResumeTime is specified, then the Time component of the ResumeTime structure is set = 0.

QUERY_SIZE: This constant shall define the size of the Results array structure.

8.1.5 Structure – STREAM

This structure defines the control parameters for a Stream. The format of the STREAM data structure is shown in figure 7.

StreamType: This variable shall identify the type of Stream defined by the STREAM structure.

Four StreamType values are defined; Asynchronous Stream (ST_Async), Available Bit Rate Stream (ST_ABR), Real–Time Stream (ST_RT) and unused (ST_Void).

StreamID: This variable shall identify the Stream. This variable is set by the Stream Driver upon successful completion of a StreamPlay call (see 8.2.4).

```
typedef struct {
  SINT StreamType;
  SINT StreamID;
  SINT Priority;
  SINT Preemption;
  SINT ClassOfService;
  SINT StreamConsumerID;
  SINT StreamProducerID;
  SINT StreamProducerTag;
  SINT StartTimeSlot;
  SINT StopTimeSlot;
  SINT PauseTimeSlot;
  SINT ResumeTimeSlot;
  union {
    SINT RTStreamType;
    struct {
      SINT Jitter;
      SINT Bandwidth;
      } RT;
    struct {
      SINT NoOfSequences;
      SINT TotalNoOfDataFrames;
      } Async;
    } Parameters;
  STIME StartTime;
  STIME StopTime;
  STIME PauseTime;
  STIME ResumeTime;
} STREAM;
```

Figure 7 – STREAM structure

Priority: This variable shall identify the priority to be used for the initial establishment of the Stream (see 5.3.6.1.2, 5.3.6.1.3, 5.3.6.1.4, 5.3.6.1.5 and 5.3.6.1.6).

Preemption: This variable shall identify whether or not to use preemption when establishing the Stream.

Two values are defined for this variable; TRUE (preemption) and FALSE (no preemption).

This variable shall only be meaningful for Available Bit Rate and Real–Time Streams.

ClassOfService: This variable shall identify the Class of Service to be used on a Stream.

Two values are defined for this variable; 2 (Class 2) and 3 (Class 3).

StreamConsumerID: This variable shall identify the destination of the Stream.

StreamProducerID: This variable shall identify the source of the Stream.

StreamProducerTag: This variable shall identify, to the Stream Producer, a specific piece of Audio Visual material or a specific Audio Visual source.

This variable shall only be meaningful for Available Bit Rate and Real–Time Streams.

StartTimeSlot: This variable shall identify the Time Slot at which Stream transmission is to begin.

StopTimeSlot: This variable shall identify the Time Slot a which Stream transmission is to cease.

This variable shall only be meaningful for Available Bit Rate and Real–Time Streams.

PauseTimeSlot: This variable shall identify the Time Slot at which Real–Time transmission is to be suspended.

This variable shall only be meaningful for Real–Time Streams.

ResumeTimeSlot: This variable shall identify the Time Slot at which Real–Time transmission is to be resumed.

This variable shall only be meaningful for Real–Time Streams.

Parameters.RTStreamType: This variable shall identify a Real–Time Stream with predefined Quality of Service parameters.

This variable shall only be meaningful for Real–Time Streams.

The following values are defined for the Parameters.RTStreamType variable:

- *0:* Undefined;
- *1:* a 601 type 270 Mb/s Stream (s_270);
- *2:* a 601 type 360 Mb/s Stream NTSC (NTSC360);
- *3:* a 601 type 360 Mb/s Stream PAL (PAL360);
- *4–127:* Reserved;
- *Others:* Invalid (see Parameters.RT.Jitter).

Parameters.RT.Jitter: This variable shall identify the largest amount of scheduling jitter allowed on a Real–Time Stream, in units of 1/4 294 967 296 second.

This variable shall only be meaningful for Real–Time Stream which are not defined by a valid Parameters.RTStreamType.

The value of this parameter shall be limited to the following values:

$$6 \times 2^{22} \geq \text{Jitter} \geq 2^{22} \quad @\ 265{,}625\ \text{MBaud}$$

$$6 \times 2^{21} \geq \text{Jitter} \geq 2^{21} \quad @\ 531{,}25\ \text{MBaud}$$

$$6 \times 2^{20} \geq \text{Jitter} \geq 2^{20} \quad @\ 1{,}062\ 5\ \text{GBaud}$$

$$6 \times 2^{19} \geq \text{Jitter} \geq 2^{19} \quad @\ 2{,}125\ \text{GBaud}$$

$$6 \times 2^{18} \geq \text{Jitter} \geq 2^{18} \quad @\ 4{,}25\ \text{GBaud}$$

Parameters.RT.Bandwidth: This variable shall define the bandwidth, in bits per second, for a Real–Time Stream.

This variable shall only be meaningful for Real–Time Streams.

The value of the parameter shall be limited to the following values:

$$\text{Bandwidth (b/s)} \begin{cases} 44\ 857\ 000 & @\ 531{,}25\ \text{MBaud} \\ 41\ 850\ 000 & @\ 1{,}062\ 5\ \text{GBau} \\ 35\ 681\ 000 & @\ 2{,}125\ \text{GBaud} \\ 23\ 343\ 000 & @\ 4{,}25\ \text{GBaud} \end{cases}$$

Parameters.Async.NoOfSequences: This variable shall identify the number of Sequences to be sent in a single Asynchronous Time Slot.

This variable shall only be meaningful for Asynchronous Streams.

Parameters.Async.TotalNoOfDataFrames:

This variable shall identify the number of Data Frames to be sent by the Stream Consumer in a single Asynchronous Time Slot.

This variable shall only be meaningful for Asynchronous Streams.

StartTime: This STIME structure shall identify the time at which Stream transmission is to begin.

StopTime: This STIME structure shall identify the time at which Stream transmission is to terminate.

This variable shall only be meaningful for Available Bit Rate and Real–Time Streams.

PauseTime: This STIME structure shall identify the time at which Real–Time Stream transmission is to be suspended.

This variable shall only be meaningful for Real–Time Streams.

ResumeTime: This STIME structure shall identify the time at which Real–Time Stream transmission is to resume.

This variable shall only be meaningful for Real–Time Streams.

8.1.6 Structure – Stream_Capabilities

This structure define the Stream capabilities for a specific Port. The format of the Stream_Capabilities data structure is shown in figure 8.

```
typedef struct {
  SINT PortID;
  SINT DataRate;
  Stream_Producer StreamProducer;
  Stream_Consumer StreamConsumer;
} Stream_Capabilities;
```

Figure 8 – Stream_Capabilities structure

PortID: This variable shall identify the Port whose Stream capabilities are captured in the structure.

DataRate: The Data Rate of the Port specified by the PortID variable. The following values are defined:

- *0:* Undefined;
- *1:* 1 062,5 MBaud;
- *2:* 2 125 MBaud;
- *4:* 4 250 MBaud;
- *Others:* Reserved.

StreamProducer: This structure shall hold the Stream capabilities of the Port, specified in the PortID variable, as a Stream Producer.

StreamConsumer: This structure shall hold the Stream capabilities of the Port, specified by the PortID variable, as a Stream Consumer.

8.1.7 Structure – Stream_Consumer

This structure define the capabilities of a Stream Consumer. The format of the Stream_Consumer data structure is shown in figure 9.

Async.MaxSeq: This variable shall define the largest number of Sequences a Stream Consumer is able to accept in a single Asynchronous Time Slot. (See 5.3.1 ’Max. number of Sequences for Asynchronous Streams’)

Async.MaxFrame: This variable shall define the largest number of Data Frames, across all Sequences, a Stream Consumer is able to accept in a single Asynchronous Time Slot. (See 5.3.1 ’Max. number of Data frames for Asynchronous Streams’)

Async.Class2.MaxSeq: This variable shall define the largest number of Class 2 Sequences a Stream Consumer is able to accept in a single Asynchronous Time Slot. (See 5.3.1 ’Max. number of Class 2 Sequences for Asynchronous Streams’)

Async.Class2.MaxFrame: This variable shall define the largest number of Class 2 Data Frames, across all Sequences, a Stream Consumer is able to accept in a single Asynchronous Time Slot. (See 5.3.1 ’Max. number of Class 2 Data frames for Asynchronous Streams’)

Async.Class2.MaxRX_IDAssign: This variable shall define the longest time, in fractions of a second (1/4 294 967 296 s), a Stream Consumer uses to assign an RX_ID to a Sequence received in an Asynchronous Time Slot. (See 5.3.1 ’Max. RX_ID assignment time for Class 2 Asynchronous Stream’)

Async.Class2.MaxInitAckLatency: This variable shall define the longest time, in fractions of a second (1/4 294 967 296 s), a Stream Consumer uses before it transmits the first ACK for a Sequence received in an Asynchronous Time Slot. (See 5.3.1 ’Max. latency for Class 2 initial ACK’)

Async.Class2.MaxFinalAckLatency: This variable shall define the longest time, in fractions of a second (1/4 294 967 296 s), a Stream Consumer uses before it transmits the final ACK for a Sequence received in an Asynchronous Time Slot. (See 5.3.1 ’Max. latency for Class 2 final ACK’)

```
typedef struct {
  struct {
    SINT MaxSeq;
    SINT MaxFrame;
    struct {
      SINT MaxSeq;
      SINT MaxFrame;
      SINT MaxRX_IDAssign;
      SINT MaxInitAckLatency;
      SINT MaxFinalAckLatency;
      } Class2;
    struct {
      SINT MaxSeq;
      SINT MaxFrame;
      } Class3;
    } Async;
  SINT MaxStream;
  SINT MaxTimeSlot;
  struct {
    SINT MaxStream;
    struct {
      SINT MaxStream;
      SINT MaxSeq;
      SINT MaxRX_IDAssign;
      SINT MaxInitAckLatency;
      SINT MaxFinalAckLatency;
      } Class2;
    struct {
      SINT MaxStream;
      SINT MaxSeq;
      } Class3;
    } ABR;
  struct {
    SINT MaxStream;
    struct {
      SINT MaxStream;
      SINT MaxTimeSlot;
      SINT MaxTimeSlotPerStream;
      SINT MaxSeq;
      SINT MaxRX_IDAssign;
      SINT MaxInitAckLatency;
      SINT MaxFinalAckLatency;
      } Class2;
    struct {
      SINT MaxStream;
      SINT MaxTimeSlot;
      SINT MaxTimeSlotPerStream;
      SINT MaxSeq;
      } Class3;
  } RT;
} Stream_Consumer;
```

Figure 9 – Stream_Consumer structure

Async.Class3.MaxSeq: This variable shall define the largest number of Class 3 Sequences a Stream Consumer is able to accept in a single Asynchronous Time Slot. (See 5.3.1 ’Max. number of Class 3 Sequences for Asynchronous Streams’)

Async.Class3.MaxFrame: This variable shall define the largest number of Class 3 Data Frames, across all Sequences, a Stream Consumer is able to accept in a single Asynchronous Time Slot. (See 5.3.1 ’Max. number of Class 3 Data frames for Asynchronous Streams’)

MaxStream: This variable shall define the largest combined number of Available Bit Rate and Real–Time Streams a Stream Consumer is able to accept. (See 5.3.1 ’SC max. Streams’)

MaxTimeSlot: This variable shall define the largest number of Time Slots per second, that a Stream Consumer is able to support across all Available Bit Rate and Real–Time Streams. (See 5.3.1 ’SP max. aggregate TSs per Stream’)

ABR.MaxStream: This variable shall define the largest number of Available Bit Rate Streams a Stream Consumer is able to accept. (See 5.3.1 ’SC max. ABR Streams’)

ABR.Class2.MaxStream: This variable shall define the largest number of Class 2 Available Bit Rate Streams a Stream Consumer is able to accept. (See 5.3.1 ’SC max. Class 2 ABR Streams’)

ABR.Class2.MaxSeq: This variable shall define the largest number of Class 2 Sequences a Stream Consumer is able to accept in a single Time Slot for an Available Bit Rate Stream. (See 5.3.1 ’SC max. Class 2 Sequences per ABR Stream TS’)

ABR.Class2.MaxRX_IDAssign: This variable shall define the longest time, in fractions of a second (1/4 294 967 296 s), a Stream Consumer uses to assign an RX_ID to a Sequence received in an Available Bit Rate Stream. (See 5.3.1 ’Max. RX_ID assignment time for Class 2 ABR Stream’)

ABR.Class2.MaxInitAckLatency: This variable shall define the longest time, in fractions of a second (1/4 294 967 296 s), a Stream Consumer uses before it transmits the first ACK for a Sequence received in an Available Bit Rate

Stream. (See 5.3.1 ’Max. latency for Class 2 initial ABR ACK’)

ABR.Class2.MaxFinalAckLatency: This variable shall define the longest time, in fractions of a second (1/4 294 967 296 s), a Stream Consumer uses before it transmits the final ACK for a Sequence received in an Available Bit Rate Stream. (See 5.3.1 ’Max. latency for Class 2 final ABR ACK’)

ABR.Class3.MaxStream: This variable shall define the largest number of Class 3 Available Bit Rate Streams a Stream Consumer is able to accept. (See 5.3.1 ’SC max. Class 3 ABR Streams’)

ABR.Class3.MaxSeq: This variable shall define the largest number of Class 3 Sequences a Stream Consumer is able to accept in a single Time Slot for an Available Bit Rate Stream. (See 5.3.1 ’SC max. Class 3 Sequences per ABR Stream TS’)

RT.MaxStream: This variable shall define the largest number of Real–Time Streams a Stream Consumer is able to accept. (See 5.3.1 ’SC max. RT Streams’)

RT.Class2.MaxStream: This variable shall define the largest number of Class 2 Real–Time Streams a Stream Consumer is able to accept. (See 5.3.1 ’SC max. Class 2 RT Streams’)

RT.Class2.MaxTimeSlot: This variable shall define the largest number of Time Slot per second, that a Stream Consumer is able to accept across all Class 2 Real–Time Streams. (See 5.3.1 ’SC max. aggregate Class 2 TSs per RT Stream’)

RT.Class2.MaxTimeSlotPerStream: This variable shall define the largest number of Time Slots per second, that a Stream Consumer is able to accept for a single Class 2 Real–Time Stream. (See 5.3.1 ’SC max. Class 2 TSs per RT Stream’)

RT.Class2.MaxSeq: This variable shall define the largest number of Class 2 Sequences a Stream Consumer is able to accept in a single Time Slot for a Real–Time Stream. (See 5.3.1 ’SC max. Class 2 Sequences per RT Stream TS’)

RT.Class2.MaxRX_IDAssign: This variable shall define the longest time, in fractions of a second (1/4 294 967 296 s), a Stream Consumer uses to assign an RX_ID to a Sequence received in a Real–Time Stream. (See 5.3.1 ’Max. RX_ID assignment time for Class 2 RT Stream’)

RT.Class2.MaxInitAckLatency: This variable shall define the longest time, in fractions of a second (1/4 294 967 296 s), a Stream Consumer uses before it transmits the first ACK for a Sequence received in a Real–Time Stream. (See 5.3.1 ’Max. latency for Class 2 initial RT ACK’)

RT.Class2.MaxFinalAckLatency: This variable shall define the longest time, in fractions of a second (1/4 294 967 296 s), a Stream Consumer uses before it transmits the final ACK for a Sequence received in a Real–Time Stream. (See 5.3.1 ’Max. latency for Class 2 final RT ACK’)

RT.Class3.MaxStream: This variable shall define the largest number of Class 3 Real–Time Streams a Stream Consumer is able to accept. (See 5.3.1 ’SC max. Class 3 RT Streams’)

RT.Class3.MaxTimeSlot: This variable shall define the largest number of Time Slot per second, that a Stream Consumer is able to accept across all Class 3 Real–Time Streams. (See 5.3.1 ’SC max. aggregate Class 3 TSs per RT Stream’)

RT.Class3.MaxTimeSlotPerStream: This variable shall define the largest number of Time Slots per second, that a Stream Consumer is able to accept for a single Class 3 Real–Time Stream. (See 5.3.1 ’SC max. Class 3 TSs per RT Stream’)

RT.Class3.MaxSeq: This variable shall define the largest number of Class 3 Sequences a Stream Consumer is able to accept in a single Time Slot for a Real–Time Stream. (See 5.3.1 ’SC max. Class 3 Sequences per RT Stream TS’)

8.1.8 Structure – Stream_Producer

This structure define the capabilities of a Stream Producer. The format of the Stream_Producer data structure is shown in figure 10.

MaxStream: This variable shall define the largest combined number of Available Bit Rate and Real–Time Streams a Stream Producer is able to support. (See 5.3.1 ’SP max. Streams’)

```
typedef struct {
  SINT MaxStream;
  SINT MaxTimeSlot;
  struct {
    SINT MaxStream;
    struct {
      SINT MaxStream;
      } Class2;
    struct {
      SINT MaxStream;
      } Class3;
    } ABR;
  struct {
    SINT MaxStream;
    struct {
      SINT MaxStream;
      SINT MaxTimeSlot;
      SINT MaxTimeSlotPerStream;
      } Class2;
    struct {
      SINT MaxStream;
      SINT MaxTimeSlot;
      SINT MaxTimeSlotPerStream;
      } Class3;
    } RT;
} Stream_Producer;
```

Figure 10 – Stream_Producer structure

MaxTimeSlot: This variable shall define the largest number of Time Slot per second, that a Stream Producer is able to support across all Available Bit Rate and Real–Time Streams. (See 5.3.1 ’SP max. aggregate TSs per Stream’)

ABR.MaxStream: This variable shall define the largest number of Available Bit Rate Streams a Stream Producer is able to support. (See 5.3.1 ’SP max. ABR Streams’)

ABR.Class2.MaxStream: This variable shall define the largest number of Class 2 Available Bit Rate Streams a Stream Producer is able to support. (See 5.3.1 ’SP max. Class 2 ABR Streams’)

ABR.Class3.MaxStream: This variable shall define the largest number of Class 3 Available Bit Rate Streams a Stream Producer is able to support. (See 5.3.1 ’SP max. Class 3 ABR Streams’)

RT.MaxStream: This variable shall define the largest number of Real–Time Streams a Stream Producer is able to support. (See 5.3.1 ’SP max. RT Streams’)

RT.Class2.MaxStream: This variable shall define the largest number of Class 2 Real–Time Streams a Stream Producer is able to support.

RT.Class2.MaxTimeSlot: This variable shall define the largest number of Time Slot per second, that a Stream Producer is able to support across all Class 2 Real–Time Streams. (See 5.3.1 ’SP max. aggregate Class 2 TSs per RT Stream’)

RT.Class2.MaxTimeSlotPerStream: This variable shall define the largest number of Time Slots per second, that a Stream Producer is able to support for a single Class 2 Real–Time Stream. (See 5.3.1 ’SP max. Class 2 TSs per RT Stream’)

RT.Class3.MaxStream: This variable shall define the largest number of Class 3 Real–Time Streams a Stream Producer is able to support. (See 5.3.1 ’SP max. Class 3 RT Streams’)

RT.Class3.MaxTimeSlot: This variable shall define the largest number of Time Slot per second, that a Stream Producer is able to support across all Class 3 Real–Time Streams. (See 5.3.1 ’SP max. aggregate Class 3 TSs per RT Stream’)

RT.Class3.MaxTimeSlotPerStream: This variable shall define the largest number of Time Slots per second, that a Stream Producer is able to support for a single Class 3 Real–Time Stream. (See 5.3.1 ’SP max. Class 3 TSs per RT Stream’)

8.2 API – Stream functions

The following functions are defined for the Class 2 and 3 Stream Driver:

- *StreamInit():* Prepares the Port for Stream operation and establish the basic environment;
- *StreamGetCapabilities():* Enable a Port to obtain the Stream capabilities of another Port;
- *StreamPause():* Serves to suspend transmission for an established Real–Time Stream;
- *StreamPlay():* Serves to establish an Available Bit Rate or Real–Time Stream, or to create an event for the Asynchronous Stream;

- *StreamQuery():* Serves to identify the different Available Bit Rate or Real–Time Streams using a given Stream Identifier;

- *StreamQueueProducer():* Serves to obtain, from a Stream Producer, a Stream handle (Producer Tag) for a specific piece of Audio Visual material or an Audio Visual source;

- *StreamResume():* Serves to resume transmission for an established Real–Time Stream;

- *StreamStop():* Serves to terminate an Available Bit Rate or Real–Time Stream;

- *StreamTime():* Enables determination of the current System Stream Time value;

The functions applicable to the three different Stream types are shown in table 122, presented in the normal order of usage.

Table 122 – Typical Stream function flow

<table>
<tr><th>Async</th><th>ABR</th><th>Real–Time</th></tr>
<tr><td colspan="3">StreamInit()</td></tr>
<tr><td colspan="3">StreamGetCapabilities()</td></tr>
<tr><td>–</td><td colspan="2">StreamQueueProducer()</td></tr>
<tr><td colspan="3">StreamPlay()</td></tr>
<tr><td rowspan="4">–</td><td colspan="2">StreamQuery()[1]</td></tr>
<tr><td rowspan="2">–</td><td>StreamPause()[1]</td></tr>
<tr><td>StreamResume()[1]</td></tr>
<tr><td colspan="2">StreamStop()</td></tr>
<tr><td colspan="3">Notes
1 Optional</td></tr>
</table>

8.2.1 Function – StreamInit()

The StreamInit function serves to prepare the Port for operation in an environment supporting either or both Class 2 and 3 Streams.

The StreamInit function shall perform the following, in the stated order:

a) Initialize the Port to the Offline state.

b) Prepare the Port to send and receive Asynchronous Streams, and if supported, either or both Available Bit Rate and Real–Time Streams.

c) Prepare the Port to accept Prompt and Selective Prompt Extended Link Service commands.

d) Prepare the Port to respond to the GSA Extended Link Service command, i.e. announce the Ports Stream capabilities. (See 5.3.1)

e) Activate the Port, performing the Link Initialization protocol (Offline to Online).

f) Attempt login with the Fabric, FLOGI and if successful, continue with step i, if unsuccessful return with an SERR code of ‘Unsuccessful FLOGI’.
FLOGI is successful if either the FLOGI process terminated with the Port logged into the Fabric or if the Port discovered it formed part of a Point–to–Point or Private Loop topology.

g) Find the address for the System Resource Manager, if unsuccessful return with an SERR code of ‘System Resource Manager not found’.

h) Engage the use of the Stream protocol for all future Fibre Channel transmissions. (See 5.1)

i) Register with the Name Server, see 5.1.3(f), 5.1.4(b) and 5.1.5(e), if unsuccessful return with an SERR code of ‘Name Server registration failure’.

j) Return with an SERR code of ‘No error (successful completion)’.

8.2.1.1 Syntax

The calling syntax for the StreamInit function is shown in figure 11.

InterfaceID: Serves to identify the Fibre Channel interface (Port) being enabled to operate in Stream mode[3)] .

8.2.1.2 Return value

The StreamInit function returns with one of the following SERR values:

- No error (successful completion) (see table 121);

```
SERR StreamInit(
      const SINT InterfaceID)
```

Figure 11 – StreamInit() function

– System Resource Manager not found (see table 121);

– Name Server registration failure (see table 121);

– Vendor unique (see table 121).

8.2.2 Function – StreamGetCapabilities()

The StreamGetCapabilities function serves to obtain another Ports Stream capabilities.

The StreamGetCapabilities function shall perform the following, in the stated order:

a) Validate the PortID parameter, if it is unacceptable return with an SERR code of 'Invalid Port Identifier'.

b) Obtain the DataRate parameter for the PortID, in a vendor unique manner[4] .

c) Send a GSA Extended Link Service command to the Fibre Channel address identified in the PortID variable. (See 5.3.1)

d) If an Extended Link Services Accept is received, then update both the StreamProducer and the StreamConsumer structures (see 8.1.6), and return with an SERR code of 'No error (successful completion)'.

e) In an Extended Link Service Reject is received, then return with an SERR code of 'Unable to get Stream Capabilities from the Port'[5] .

8.2.2.1 Syntax

The calling syntax for the StreamGetCapabilities function is shown in figure 12.

```
SERR StreamGetCapabilities(
      Stream_Capabilities *const x)
```

Figure 12 – StreamGetCapabilities()

x: A pointer to a Stream_Capabilities structure (see 8.1.6) with the following defined:

– *PortID:* Set to the Fibre Channel address, whose Stream Capabilities to obtain and store in the Stream_Capabilities structure (see 8.1.6).

8.2.2.2 Return value

The StreamInit function returns with one of the following SERR values:

– No error (successful completion) (see table 121);

– Invalid Port Identifier (see table 121);

– Unable to get Stream Capabilities from the Port (see table 121);

– Vendor unique (see table 121).

8.2.3 Function – StreamPause()

The StreamPause function serves to suspend, pause an established Real–Time Stream.

The StreamPause function shall perform the following, in the stated order:

a) Validate the StreamType parameter, if it is unacceptable return with an SERR code of 'Invalid Stream Type'.

b) Validate the StreamID parameter, if it is unacceptable return with an SERR code of 'Invalid Stream Identifier'.

c) Validate the StreamProducerID parameter, if it is unacceptable return with an SERR code of 'Invalid Stream Producer'.

d) Validate the PauseTime parameter, if it is unacceptable return with an SERR code of 'Invalid Pause Time'.

e) Generate and transmit an SRTS Stream request (see 5.3.6.1.8).

f) If an accept to the SRTS Stream request is received, then set the PauseTimeSlot variable as defined in the accept (see 5.3.6.3.1.11).

g) If an accept to the SRTS Stream request is received, then return with an SERR code of

---

[3] Enables the Stream Driver to access any information required to perform the tasks assigned to the StreamInit function.

[4] May be set as 'Undefined'.

[5] Transmission of the GSA Extended Link Service command may be retried if warranted by the LS_RJT reason code.

'No error (successful completion)', otherwise return the S_RJT reason code (see 5.3.6.3.2).

8.2.3.1 Syntax

The calling syntax for the StreamPause function is shown in figure 13.

```
SERR StreamPause(
      STREAM *const stream)
```

Figure 13 – StreamPause() function

stream: A pointer to a STREAM structure (see 8.1.5) with the following variables defined:

– *StreamType:* Set to Real–Time Stream Type (see 8.1.5). This variable is normally already defined by a prior call to the StreamQueueProducer function.

– *StreamID:* Set to the Stream Identifier assigned to the Stream being terminated. This variable is normally already defined by a prior call to the StreamPlay function.

– *StreamProducerID:* Set to the Fibre Channel address of the source of the Stream, the Stream Producer. This variable is normally already defined by a prior call to the StreamQueueProducer function.

– *PauseTime:* Set to the System Stream Time at which Stream transmission is to be suspended.
A PauseTime value of 'Time = 0', shall be used to mean a request for an immediate suspension of the Real–Time Stream.

8.2.3.2 Return value

The StreamPause function returns with one of the following SERR values:

– No error (successful completion) (see table 121);

– Invalid Stream Identifier (see table 54);

– Invalid Stream Producer (see table 54);

– Invalid Stream Type (see table 121);

– Invalid Pause Time (see table 121);

– Insufficient System Resource Manager resources to queue request (see table 54);

– Vendor unique (see table 121).

8.2.4 Function – StreamPlay()

The StreamPlay function serves to initiate a new Available Bit Rate or Real–Time Stream or to add an event to the Asynchronous Stream.

The StreamPlay function shall perform the following, in the stated order:

a) Validate the StreamType parameter, if it is unacceptable, then return with an SERR code of 'Invalid Stream Type'.

b) Validate the Priority parameter, if it is unacceptable, then return with an SERR code of 'Invalid Priority'.

c) Validate the ClassOfService parameter, if it is unacceptable, then return with an SERR code of 'Invalid Class of Service'.

d) Validate the StreamConsumerID parameter, if it is unacceptable, then return with an SERR code of 'Invalid Stream Consumer'.

e) Validate the StreamProducerID parameter, if it is unacceptable, then return with an SERR code of 'Invalid Stream Producer'.

f) If the StreamType is an Asynchronous Stream, then validate the:

1) Parameters.Async.NoOfSequences parameter, if it is unacceptable, then return with an SERR code of 'Invalid Number of Sequences';

2) Parameters.Async.TotalNoOfDataFrames parameter, if it is unacceptable, then return with an SERR code of 'Invalid Total Number of Data Frames'.

g) If the StreamType is either an Available Bit Rate or a Real–Time Stream, then validate the:

1) StartTime parameter, if it is unacceptable, then return with an SERR code of 'Invalid Start Time';

2) Preemption parameter, if it is unacceptable, then return with an SERR code of 'Invalid Preemption parameter';

3) StreamProducerTag parameter, if it is unacceptable, then return with an SERR code of 'Invalid Producer Tag'.

h) If the StreamType is a Real–Time Stream, then validate the:

1) Parameters.RTStreamType and Parameters.RT.Jitter parameter, if both are unacceptable, then return with an SERR code of ‘Invalid Real–Time Stream Type’, else, if only the Parameters.RTStreamType parameter was unacceptable, validate the:

i) Parameters.RT.Jitter parameter, if it is unacceptable, then return with an SERR code of ‘Invalid Jitter’;

ii) Parameters.RT.Bandwidth parameter, if it is unacceptable, then return with an SERR code of ‘Invalid Bandwidth’.

i) Generate and transmit a PRABRS, PRRTS, RAS, RABRS or RRTS Stream request, as appropriate (see 5.3.6.1.2, 5.3.6.1.3, 5.3.6.1.4, 5.3.6.1.5 and 5.3.6.1.6).

j) If an accept to the PRABRS, PRRTS, RAS, RABRS or RRTS Stream request is received, then set the StartTimeSlot variable as defined in the accept (see 5.3.6.3.1).

k) If an accept to the PRABRS, PRRTS, RAS, RABRS or RRTS Stream request is received, then return with an SERR code of ‘No error (successful completion)’, otherwise return the S_RJT reason code (see 5.3.6.3.2).

8.2.4.1 Syntax

The calling syntax for the StreamPlay function is shown in figure 14.

```
SERR StreamPlay(
      STREAM *const stream,
      Stream_Producer *const SP,
      Stream_Consumer *const SC)
```

Figure 14 – StreamPlay() function

stream: A pointer to a STREAM structure (see 8.1.5) with the following variables defined:

– *StreamType:* Set to either Asynchronous, Available Bit Rate or Real–Time Stream Type (see 8.1.5).
This variable is normally already defined for Available Bit Rate and Real–Time Streams, by a prior call to the StreamQueueProducer function.

– *Priority:* Set to the desired priority level (see 8.1.5).

– *Preemption:* Set to enable or disable the use of Preemption (see 8.1.5, 5.3.6.1.2 and 5.3.6.1.3).
This variable shall only have meaning for Available Bit Rate and Real–Time Streams.

– *ClassOfService:* Set to the Class of Service to be used by the Stream (see 8.1.5).

– *StreamConsumerID:* Set to the Fibre Channel address of the destination of the Stream, the Stream Consumer.

– *StreamProducerID:* Set to the Fibre Channel address of the source of the Stream, the Stream Producer.
This variable is normally already defined for Available Bit Rate and Real–Time Streams, by a prior call to the StreamQueueProducer function.

– *StreamProducerTag:* Set to identify a specific piece of Audio Visual material or specific Audio Visual source in the Stream Producer.
This variable is normally already defined for Available Bit Rate and Real–Time Streams, by a prior call to the StreamQueueProducer function.

– *StartTime:* Set to the System Stream Time at which Stream transmission is to be begin.
A StartTime value of ‘Time = 0’, shall be used to mean an immediate start of the requested Stream.
This variable shall only be meaningful for Available Bit Rate and Real–Time Streams.

– *Parameters.Async.TotalNoOfDataFrames:* Set to the total number of Data Frames to be sent, across all Sequences, in a single Asynchronous Time Slot (see 8.1.5).
This variable shall only have meaning for Asynchronous Streams.

– *Parameters.Async.NoOfSequences:* Set to the number of Sequences to be sent in a single Asynchronous Time Slot (see 8.1.5).
This variable shall only have meaning for Asynchronous Streams.

– *Parameters.RTStreamType:* Set to the predefined Real–Time Stream type desired

(see 8.1.5).
This variable shall only have meaning for Real–Time Streams which are using predefined Quality of Service parameters.

– *Parameters.RT.Jitter:* Set to the desired maximum Real–Time source jitter limit (see 8.1.5).
This variable shall only have meaning for Real–Time Streams which are not using predefined Quality of Service parameters.

– *Parameters.RT.Bandwidth:* Set to the desired Real–Time bandwidth limit (see 8.1.5).
This variable shall only have meaning for Real–Time Streams which are not using predefined Quality of Service parameters.

SP: A pointer to a Stream_Producer structure (see 8.1.8) holding the Stream capabilities of the Stream Producer.

SC: A pointer to a Stream_Consumer structure (see 8.1.7) holding the Stream capabilities of the Stream Consumer.

8.2.4.2 Return value

The StreamPlay function returns with one of the following SERR values:

- No error (successful completion) (see table 121);
- Invalid Stream Type (see table 121);
- Invalid Priority (see table 54);
- Invalid Class of Service (see table 54);
- Invalid Stream Consumer (see table 54);
- Invalid Stream Producer (see table 54);
- Invalid Number of Sequences (see table 54);
- Invalid Total Number of Data Frames (see table 54);
- Invalid Start Time (see table 121);
- Invalid Preemption parameter (see table 121);
- Invalid Producer Tag (see table 54);
- Invalid Real–Time Stream Type (see table 54);
- Invalid Jitter (see table 54);
- Invalid Bandwidth (see table 54);
- Stream Producer unable to source another ABR or Real–Time Stream (see table 54);
- Stream Producer unable to source requested bandwidth (see table 54);
- Stream Consumer unable to accept another ABR or Real–Time Stream (see table 54);
- Stream Consumer unable to accept requested bandwidth (see table 54);
- Insufficient System bandwidth to honor request (see table 54);
- Insufficient System Resource Manager resources to queue request (see table 54);
- Vendor unique (see table 121).

8.2.5 Function – StreamQuery()

The StreamQuery function serves to obtain detailed information of the individual Streams assigned to a given Stream Identifier.

The StreamQuery function shall perform the following, in the stated order:

a) Validate the StreamID parameter, if it is unacceptable, then return with an SERR code of ‘Invalid Stream Identifier’.

b) Validate the Length parameter, if it is unacceptable, then return with an SERR code of ‘Invalid size of Query Results array’.

c) Send a Query Extended Link Service command to the System Resource Manager. (See 5.3.4)

d) If an Extended Link Services Accept is received, then update the Results array, limited by either the size of the Results array or the number of Responses in the Accept, and set Length = ‘number of entries in the Results array’.
If the number Responses (see 5.3.4) exceed the size of the Results array, then return with an SERR code of ‘Overflow’, otherwise return with an SERR code of ‘No error (successful completion)’.

e) In an Extended Link Service Reject is received, then set Length = 0 and return with an SERR code of ‘No error (successful completion)’.

8.2.5.1 Syntax

The calling syntax for the StreamQuery function is shown in figure 15.

```
SERR StreamQuery( QUERY *const Q)
```

Figure 15 – StreamQuery() function

Q: A pointer to a QUERY structure at which, a snapshot of the Streams associated with a specific Stream Identifier may be stored. The following variables shall be defined for the structure:

- *StreamID:* Set to the Stream Identifier whose constituent Streams are desired to be found.
- *Length:* Set = QUERY_SIZE (see 8.1.4), i.e. the maximum size of the Results array.

8.2.5.2 Return value

The StreamQuery function returns with one of the following SERR values:

- No error (successful completion) (see table 121);
- Invalid Stream Identifier (see table 54);
- Vendor unique (see table 121).

8.2.6 Function – StreamQueueProducer()

The StreamQueueProducer function serves to obtain a Stream handle (Producer Tag) from a Stream Producer for a specific Audio Visual piece of material or for a specific Audio Visual Stream source.

The StreamQueueProducer function shall perform the following, in the stated order:

a) Validate the StreamType parameter, if it is unacceptable, then return with an SERR code of ‘Invalid Stream Type’.

b) Validate the StreamProducerID parameter, if it is unacceptable return with an SERR code of ‘Invalid Stream Producer’.

c) Validate the VendorUniqueSpecifier, if it is unacceptable, then return with an SERR code of ‘Invalid Audio Visual source specification’.

d) Request a Stream handle, the Producer Tag, from the Stream Producer in a vendor unique manner. If this fails then return with an SERR code of ‘Unable to assign Producer Tag’, otherwise set the StreamProducerTag and return with an SERR code of ‘No error (successful completion)’.

8.2.6.1 Syntax

The calling syntax for the StreamQueueProducer function is shown in figure 16.

```
SERR StreamQueueProducer(
      STREAM *const stream,
      VendorUniqueSpecifier)
```

Figure 16 – StreamQueueProducer() function

stream: A pointer to a STREAM structure (see 8.1.5) with the following variables defined:

- *StreamType:* Set to either Available Bit Rate or Real–Time Stream Type (see 8.1.5).
- *StreamProducerID:* Set to the Fibre Channel address of the source of the Stream, the Stream Producer.

VendorUniqueSpecifier: Set to identify, in a vendor unique manner, a specific piece of Audio Visual material to be accessed, such as a file, or to identify a specific Audio Visual source, such as a camera.

8.2.6.2 Return value

The StreamQueueProducer function returns with one of the following SERR values:

- No error (successful completion) (see table 121);
- Invalid Audio Visual source specification (see table 121);
- Unable to assign Producer Tag (see table 121);
- Invalid Stream Producer (see table 54);
- Vendor unique (see table 121).

8.2.7 Function – StreamResume()

The StreamResume function serves to resume, restart, an established Real–Time Stream.

The StreamResume function shall perform the following, in the stated order:

a) Validate the StreamType parameter, if it is unacceptable return with an SERR code of 'Invalid Stream Type'.

b) Validate the StreamID parameter, if it is unacceptable return with an SERR code of 'Invalid Stream Identifier'.

c) Validate the StreamProducerID parameter, if it is unacceptable return with an SERR code of 'Invalid Stream Producer'.

d) Validate the ResumeTime parameter, if it is unacceptable return with an SERR code of 'Invalid Resume Time'.

e) Generate and transmit an RSRTS Stream request (see 5.3.6.1.7).

f) If an accept to the RSRTS Stream request is received, then set the ResumeTimeSlot variable as defined in the accept (see 5.3.6.3.1.9).

g) If an accept to the RSRTS Stream request is received, then return with an SERR code of 'No error (successful completion)', otherwise return the S_RJT reason code (see 5.3.6.3.2).

8.2.7.1 Syntax

The calling syntax for the StreamResume function is shown in figure 17.

```
SERR StreamResume(
      STREAM *const stream)
```

Figure 17 – StreamResume() function

stream: A pointer to a STREAM structure (see 8.1.5) with the following variables defined:

– *StreamType:* Set to Real–Time Stream Type (see 8.1.5). This variable is normally already defined by a prior call to the StreamQueueProducer function.

– *StreamID:* Set to the Stream Identifier assigned to the Stream being terminated. This variable is normally already defined by a prior call to the StreamPlay function.

– *StreamProducerID:* Set to the Fibre Channel address of the source of the Stream, the Stream Producer. This variable is normally already defined by a prior call to the StreamQueueProducer function.

– *ResumeTime:* Set to the System Stream Time at which Stream transmission is to be resumed.
A ResumeTime value of 'Time = 0', shall be used to mean a request for an immediate resumption of transmission on the Real–Time Stream.

8.2.7.2 Return value

The StreamResume function returns with one of the following SERR values:

– No error (successful completion) (see table 121);

– Invalid Stream Identifier (see table 54);

– Invalid Stream Producer (see table 54);

– Invalid Stream Type (see table 121);

– Invalid Resume Time (see table 121);

– Insufficient System Resource Manager resources to queue request (see table 54);

– Vendor unique (see table 121).

8.2.8 Function – StreamStop()

The StreamStop function serves to terminate an established Available Bit Rate or Real–Time Stream.

The StreamStop function shall perform the following, in the stated order:

a) Validate the StreamType parameter, if it is unacceptable return with an SERR code of 'Invalid Stream Type'.

b) Validate the StreamID parameter, if it is unacceptable return with an SERR code of 'Invalid Stream Identifier'.

c) Validate the StreamProducerID parameter, if it is unacceptable return with an SERR code of 'Invalid Stream Producer'.

d) Validate the StopTime parameter, if it is unacceptable return with an SERR code of 'Invalid Stop Time'.

e) Generate and transmit a TABRS or TRTS Stream request (see 5.3.6.1.9 and 5.3.6.1.10).

f) If an accept to the TABRS or TRTS Stream request is received, then set the

StopTimeSlot variable as defined in the accept (see 5.3.6.3.1).

g) If an accept to the TABRS or TRTS Stream request is received, then return with an SERR code of ‘No error (successful completion)’, otherwise return the S_RJT reason code (see 5.3.6.3.2).

8.2.8.1 Syntax

The syntax for the StreamStop function is shown in figure 18.

```
SERR StreamStop(
      STREAM *const stream)
```

Figure 18 – StreamStop() function

stream: A pointer to a STREAM structure (see 8.1.5) with the following variables defined:

– *StreamType:* Set to either Available Bit Rate or Real–Time Stream Type (see 8.1.5). This variable is normally already defined by a prior call to the StreamQueueProducer function.

– *StreamID:* Set to the Stream Identifier assigned to the Stream being terminated. This variable is normally already defined by a prior call to the StreamPlay function.

– *StreamProducerID:* Set to the Fibre Channel address of the source of the Stream, the Stream Producer. This variable is normally already defined by a prior call to the StreamQueueProducer function.

– *StopTime:* Set to the System Stream Time at which Stream transmission is to be terminated.
A StopTime value of ‘Time = 0’, shall be used to mean a request for an immediate termination of the Stream.

8.2.8.2 Return value

The StreamStop function returns with one of the following SERR values:

– No error (successful completion) (see table 121);

– Invalid Stream Identifier (see table 54);

– Invalid Stream Producer (see table 54);

– Invalid Stream Type (see table 121), if neither Available Bit Rate or Real–Time Stream is defined;

– Invalid Stop Time (see table 121);

– Insufficient System Resource Manager resources to queue request (see table 54);

– Vendor unique (see table 121).

8.2.9 Function – StreamTime()

The StreamTime function serves to set a Stream Time structure to the current value of System Stream Time.

8.2.9.1 Syntax

The calling syntax for the StreamTime function is shown in figure 19.

```
SERR StreamTime(
      STIME *const stime)
```

Figure 19 – StreamTime() function

stime: A pointer to a Stream Time structure, as shown in 8.1.3, into which the current System Stream Time shall be stored.

8.2.9.2 Return value

The StreamTime function returns with one of the following SERR values:

– No error (successful completion) (see table 121);

– System Stream Time unavailable (see table 121), if the Stream Diver is not synchronized with the System Resource Manager;

– Vendor unique (see table 121).

9 Fast File Transfer (FFT) Protocol

9.1 FFT Protocol Overview

The Fast File Transfer (FFT) Protocol on Fibre Channel consists of several interfaces (See figure 20). At the high level there is a simple application interface to allow the user to select the source and destination files and the remote host name. There are two lower level interfaces. The first is the high speed interface meant for data movement which uses the SCSI-3 Fibre Channel Protocol (FCP) and specifically the SCSI commands. The second low level interface is a TCP-IP connection which is used to setup a file transfer and tear down the file transfer. Over the TCP-IP connection the file is requested and status is checked as well as the initiator/client learning characteristics of the remote target/server which will be used when the file transfer is occurring over Fibre Channel. The simplified connectivity is shown in figure 21

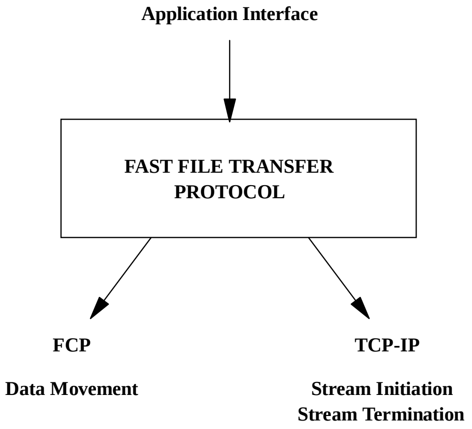

Figure 20 – Fast File Transfer Protocol High Level Interfaces

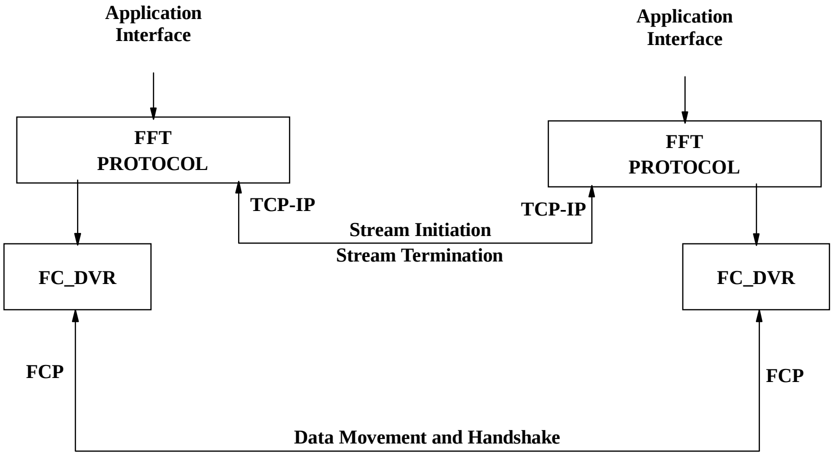

Figure 21 – FFT Protocol Connectivity

9.2 Application Interface

The application interface for the FFT Protocol is used to select the source and destination files to be transferred using the FFT Protocol.

- fftcp <rhost>:<rfile> <lfile>

This example copies a file named rfile on a remote host named rhost to a local file called lfile.

9.3 FFT Control over TCP-IP

The TCP-IP connection is used for initiation and termination of the stream. During the initialization phase the initiator/client will, using the VtpMsgOpen format and the message ID of VTP_OPEN_REQ, request a stream to be opened by the target/server. The target/server will respond using the VtpMsgOpen format and the message ID VTP_OPEN_REPLY. After the buffers have been filled on the target/server it will respond with a number of buffers, the buffer size, and the base address of the buffers using the VTP_OPEN_REQ. This base address with an offset will be the location where the initiator/client will be requesting SCSI reads from using the SCSI FCP. At the point the SCSI protocol will commence for the given stream. Meanwhile, the initiator/client will periodically 'ping' the target/server as a timeout mechanism.

Table 123 – FFT Message Ids

Dec value	Description
-1	VTP_UNKNOWN
0	VTP_OPEN_REQ
1	VTP_OPEN_REPLY
2	VTP_CLOSE_REQ
3	VTP_CLOSE_REPLY
4	VTP_PRIME_REQ
5	VTP_PRIME_REPLY
6-8	Reserved
9	VTP_COPY_CLOSE_REQ
10	VTP_COPY_CLOSE_REPLY
11-49	Reserved
50	VTP_PING_REQ
51	VTP_PING_REPLY
52-99	Reserved

The are two structures of messages exchanged between client and server over TCP-IP. The first structure VtpMsgOpen is the message format for communication between local and remote VTP process during the opening of the stream and is only used for message IDs of VTP_OPEN_REQ and VTP_OPEN_REPLY. The second communication structure is called VtpMsg. This structure is used for all other communications between the client and the server. The two structures are shown below.

9.3.1 VtpMsgOpen

```
typedef struct {
    /* VTP protocol revision */
  int version;
    /* Message ID */
  int mid;
    /* Stream Handle */
  int strmHd;

    /* file info */
  char svrName[VTP_MAX_NAME_LEN+1;
  char fileName[VTP_MAX_NAME_LEN+1]
;
  VtpOffset fileSize;
  int bandWidth;
  int mode;
  int svcType;
    /* Number of remote side stream
buffers required */
  int mgcBufNum;
    /* Size in byte of each stream
buffer */
  int mgcBufSize;
    /* STD_TRACK or FT_TRACK */
  int trackType;

/* response info */
  int result;
  int mgcBaseOffset;
    /* File System Buffer Size */
  int mdsBufSize;
} VtpMsgOpen;
```

9.3.2 VtpMsg

```
typedef struct {
    /* message type, CONTROL or DA-
TA... necessary? */
  int mtype;
    /* Message ID */
  int mid;
    /* Stream Handle */
  int strmHd;
    /* Argument or result */
  int info;
    /* buffer information */
    /* buffer address */
  char* buf;
    /* data length in buffer */
  int len;
    /* buffer size */
  int size;
    /* logical block number */
  int logicalBlock;
    /* delay before disk I/O */
  int delayTime;
    /* user data */
  void* cargo;
    /* playTrack or frameTableTrack
*/
  int trackType;
    /* reserved for now */
  int ext;
} VtpMsg;
```

9.4 FFT Connection using TCP/IP

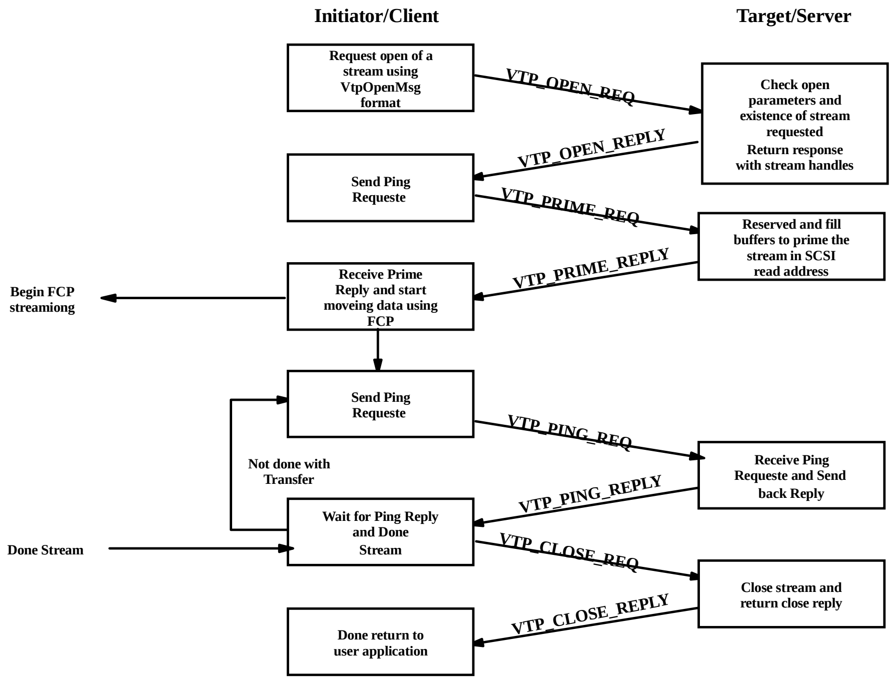
Initiator/Client
Target/Server
Request open of a stream using VtpOpenMsg format
VTP_OPEN_REQ
Check open parameters and existence of stream requested
Return response with stream handles
VTP_OPEN_REPLY
Send Ping Requeste
VTP_PRIME_REQ
Reserved and fill buffers to prime the stream in SCSI read address
VTP_PRIME_REPLY
Begin FCP streamiong
Receive Prime Reply and start moveing data using FCP
Send Ping Requeste
VTP_PING_REQ
Receive Ping Requeste and Send back Reply
Not done with Transfer
VTP_PING_REPLY
Wait for Ping Reply and Done Stream
Done Stream
VTP_CLOSE_REQ
Close stream and return close reply
VTP_CLOSE_REPLY
Done return to user application

Figure 22 – TCP/IP FFT Protocol Flow Chart

9.5 FFT data movement using FCP

The data movement uses FCP, specifically SCSI on Fibre Channel. The normal SCSI protocol is shown below. A modification for use in the FFT Protocol is that the after the normal SCSI read or write command is finished for a given portion of the file transfer a SCSI Send Diagnostic command is sent from the receiver of the data (initiator/client in the case of a read and target/server in the case of a write). This command is used to let the provider of data know that a given SCSI read or write was successful so that the buffer of data from the source may be freed and reused to be file by the data providers system of filling buffers.

9.6 Walk through of Activities via TCP-IP and FCP

Assume we have two systems (initiator/client and target/server) in a network. Figure 24 shows the elements of the overall scenario.

We would like open a stream which is located on the target’s local disk for transfer to the initiators local disk. This means that initiator will be READing the data from target once the data is ready to be read.

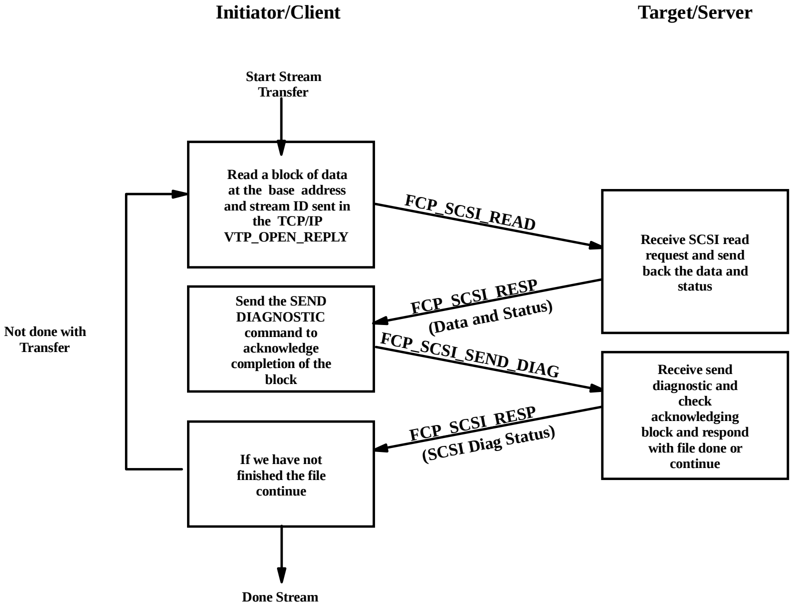
Initiator/Client
Target/Server
Start Stream Transfer
Read a block of data at the base address and stream ID sent in the TCP/IP VTP_OPEN_REPLY
FCP_SCSI_READ
Receive SCSI read request and send back the data and status
FCP_SCSI_RESP (Data and Status)
Send the SEND DIAGNOSTIC command to acknowledge completion of the block
Not done with Transfer
FCP_SCSI_SEND_DIAG
Receive send diagnostic and check acknowledging block and respond with file done or continue
FCP_SCSI_RESP (SCSI Diag Status)
If we have not finished the file continue
Done Stream

Figure 23 – FCP SCSI FFT Protocol Flow Chart for FFT Protocol

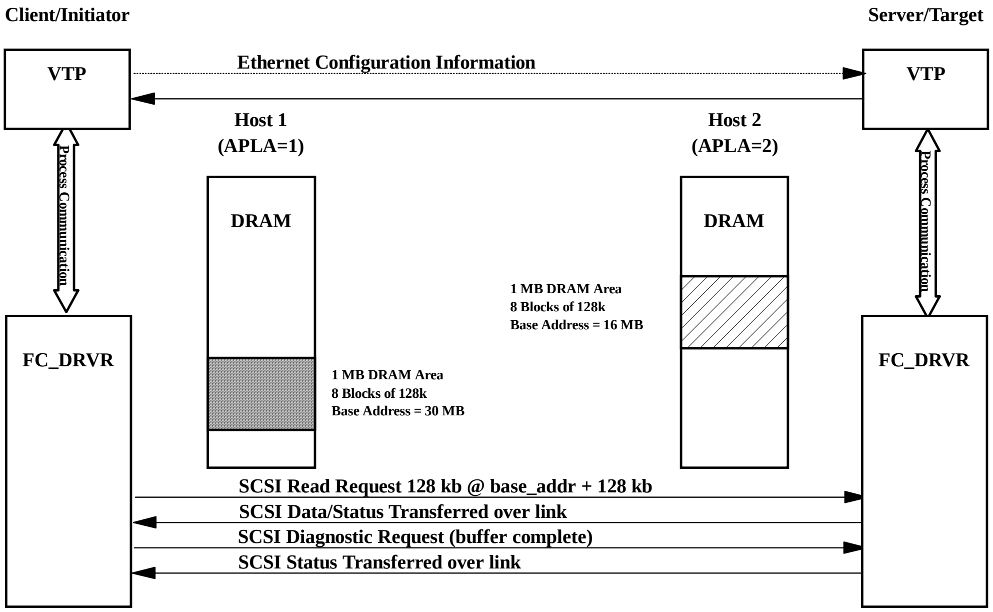

Figure 24 – Components of FFT

The initiator and target will establish over ethernet as in section 2.2 and the initiator receives the following information for example number of buffers to use (4), the base address for the streaming (10 MB), and the size of the blocks (384 k). This assumes no errors occur in establishing the connection and the requested material is available.

The prime command is issued by the initiator/client to the target/server and the buffers shown in the target in figure 25 are filled.

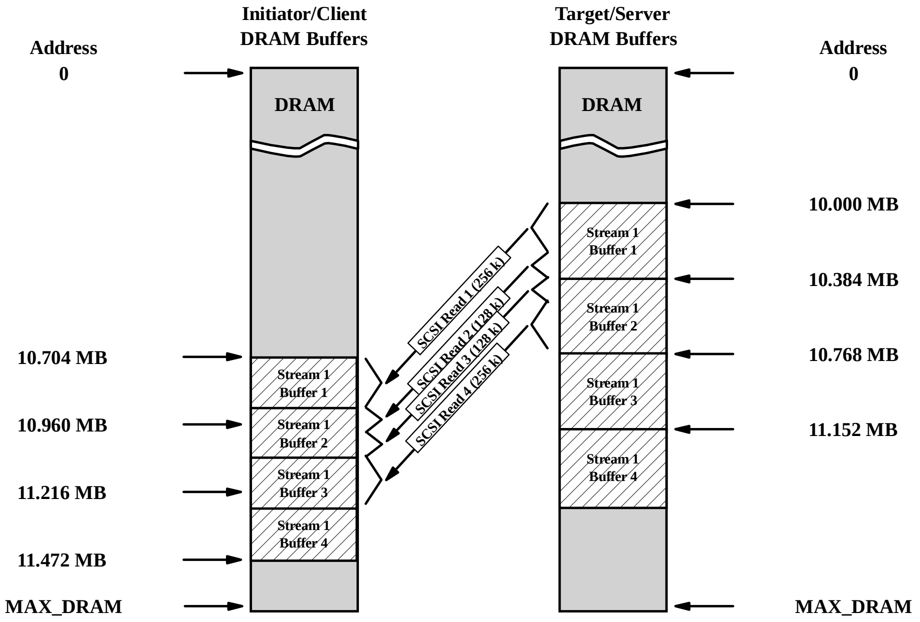

Figure 25 – DRAM Buffers involved in SCSI portion of FFT

The initiator will now do SCSI requests in the way described in section 2.3. The sizes of the requests are shown such that even for non-matched initiator and target buffer sizes data can be read by the initiator.

When a buffer on the target has moved completely across to the initiator (which may take several SCSI transfers) that buffer is freed on the target and refilled on the target as soon as possible so as to continue the value of streaming.

When a buffer on the initiator in filled that buffer is given to the initiators file system to put onto the locally created file. Once that buffer is completely moved to the local file system, the buffer is freed back to the initiators receive buffer area.

When the entire file or portion of the file has moved out of the target and into the initiator, the process is complete.

10 FC-AV Containers

10.1 Overview

The FC-AV Container system is designed to optimize the transport of video data over Fibre Channel. An essential element of this optimization is the grouping of video, audio and ancillary data into relatively large data sets to be transported as a unit. Using 4:2:2 uncompressed video as an example; the data set is of the order of one megabyte per video frame. Fibre Channel is well suited to block transmission of such a large data set, and is much more efficient when data is grouped in large blocks rather than small packets.

Fibre Channel implementations use sophisticated chip sets to “Exchange” data. At the lowest level, the Exchange is composed of frames limited to 2K bytes. In practice, these Fibre Channel Frames are not visible to any part of the system above the interface chip(s). The lowest level that is visible is referred to as a Sequence. A Sequence is an element of an Exchange based on the particular protocol used. The Container proposed in this clause maps to the Fibre Channel Sequence. This clause does not address higher level protocols required to set-up Exchanges of Sequences, but only the packaging of the data in a single Sequence. References are made to streams of data such as a stream of video frames, which are in effect a series of video frames packaged into Objects and sent in sequential Exchanges over Fibre Channel links. Mechanisms are provided to insure the Objects can be set in the correct order, and provide enough information to disseminate the data by type (Video, Audio and Ancillary data) at the lowest possible level.

An important aspect of the Container definition is the ability to utilize the Container header as a mechanism for directing Objects composed of elemental data (such as video samples or audio samples) directly to decoders with a minimum of software overhead. This direction is designed to happen at the lowest possible level for high efficiency.

More sophisticated applications transporting streams with multiple elements, and metadata may require the intervention of a higher protocol layer. An extended header is defined to address a variable number of Objects.

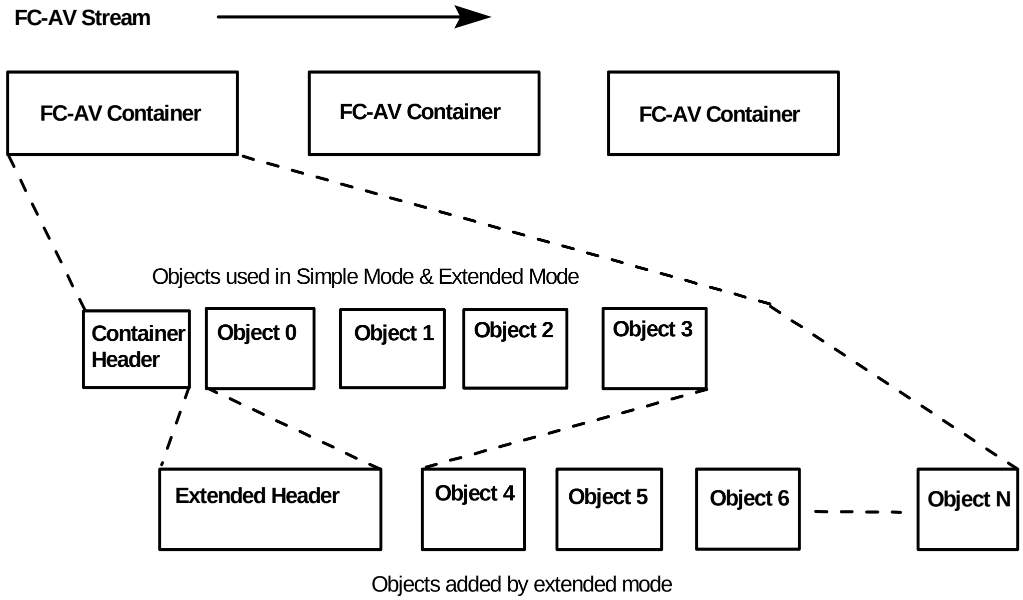

Figure 26 – The FC-AV Container System

10.2 The FC-AV Container System

A Container system (see figure 26) is proposed for transporting video or graphics with associated audio and ancillary data. Each Container is intended to hold a video frame (optionally two fields) worth of video, audio and ancillary data. Each data type is held within an Object. Simple mode supports 4 Objects of somewhat restricted types. Extended header mode supports up to 16 Objects. All Objects, except the Container header, are optional. A single Container maps exactly into a single Fibre Channel Sequence.

10.2.1 The FC-AV Container

Streaming of video data is accomplished by sequential transmission of FC-AV Containers at the nominal video frame rate. Within each Container a number of Objects exist containing video, audio and metadata elements. A fixed length header precedes data Objects. An optional extended header directly follows the fixed length header for the extended header mode. The first data Object, Object 0, directly follows the end of the header. Specific Container types are defined in the Object classification system (see clause 10.6). Two basic modes of operation are defined; Simple mode and Extended Header mode.

The Simple mode is targeted at applications desiring to send a stream of video and or audio with a minimum of overhead. It has attributes designed to be easy to implement in hardware. The Container header addresses the first 4 Objects. The simple mode supports only those 4 Objects. The Container header is of fixed length, with Object descriptions in sequence and restricted to certain data types.

The Extended Header mode provides support for more complex streams composed of a number of elements requiring additional features not included in the Simple Header Mode. Specifically the Extended Header supports the use of a variable number of Objects from 0 to 15. Physically the extended header follows the Container header, and shares the same structure. This allows the combination of the Container header and the extended header to be viewed as a single header of variable length.

10.2.2 FC-AV concepts for Containers in simple mode

Following is a list of basic concepts describing FC-AV Containers in simple mode:

a) The FC-AV Container is a proposal for the encapsulation of video, audio and graphic data. Simple mode is defined as a null extended header, thus restricting the number of Objects containing data to four. They are referenced by the location number, or Objects 0-3

b) For video applications the Container nominally represents a video frame time. Audio and ancillary data is nominally associated with the “time span” represented by the video frame. The Containers may be transmitted at Real Time that is the video frame rate, or at a faster or slower rate as described in the transmission word.

c) The simple mode Container has restricted data types for each of the 4 possible Objects. Valid Objects are:

- Object 0 - Restricted to Ancillary data types (Object type number 50h).
- Object 1 - Restricted to Audio data Object types (Object type number 4xh
- Object 2 - Restricted to certain Video data Object types (Object type number 10-2Fh)
- Object 3 - Restricted to certain Video data Object types (Object type number 10-2Fh)

d) The Container must have a Header Object. In simple mode the extended header is null. It can optionally have data in Objects 0, 1, 2 and 3.

e) The Container header is designed to have enough information to allow direct processing of the data in the Objects without having to “crack” the individual Objects for additional content information. This is intended to allow hardware processing at the Container level if desired. Compressed video types will have additional information in the video Object to facilitate decompression.

f) The Container header is in a fixed format.

g) For the transmission of faster (or slower) than Real Time a provision is included in the Transmission Type word. The approach is to send a Container for each video frame (applies to video data not compressed temporally, such as Uncompressed, JPEG and DV) allowing the protocol processing to operate the same for Real Time and faster than Real Time transmission.

10.3 FC-AV Container header structure

The Container header is a fixed length of 22 four-byte words. The organization of the header is fixed to encourage hardware decoding of header information. Each word maps to a FC transmission word (32 bits). To facilitate use in high performance computers, the header is organized to parse well into 64 bit words. All other references to “words” in this clause refer to 32 bit FC transmission words. The first 6 words of the header apply to the entire Container. The remaining 16 words are divided into 4 identically structured sections, one for each data Object. The extended header mode adds from 1 to 12 more identically structured sections to describe Objects 4-15. The extended header is of variable length, 4 words for each Object beyond Object 3.

Table 124 – FC-AV Container Header

Word	Identifier	Byte 0	Byte 1	Byte 2	Byte 3
0	Sequence Number	MSB	-number-	-number-	LSB
1	Clip ID	MSB	-clip ID-	-clip ID-	LSB
2	Container Time Stamp	MSB	-time stamp-	-time stamp-	-time stamp-
3	Container Time Stamp	-time stamp-	-time stamp-	-time stamp-	LSB
4	Transmission Type	Video Fr. Rate	Trans. Rate	Reserved	Reserved
5	Container Type	Type	# of Objects	Reserved	Sz of Ext Hdr
6	Object 0 Class	Type 01h (00h)	Reserved	Index (00h)	Index (00h)
7	Object 0 Size	MSB	-size-	-size-	LSB
8	Object 0 Offset	MSB	-offset-	-offset-	LSB
9	Object 0 O.T. Defined	O.T. Defined	O.T. Defined	O.T. Defined	O.T. Defined
10	Object 1 Class	Type 30h	Reserved	Index (xxh)	Index (xxh)
11	Object 1 Size	MSB	-size-	-size-	LSB
12	Object 1 Offset	MSB	-offset-	-offset-	LSB
13	Object 1 O.T. Defined	O.T. Defined	O.T. Defined	O.T. Defined	O.T. Defined
14	Object 2 Class	Type xxh	Reserved	Index (xxh)	Index (xxh)
15	Object 2 Size	MSB	-size-	-size-	LSB
16	Object 2 Offset	MSB	-offset-	-offset-	LSB
17	Object 2 O.T. Defined	O.T. Defined	O.T. Defined	O.T. Defined	O.T. Defined
18	Object 3 Class	Type xxh	Reserved	Index (xxh)	Index (xxh)
19	Object 3 Size	MSB	-size-	-size-	LSB
20	Object 3 Offset	MSB	-offset-	-offset-	LSB
21	Object 3 O.T. Defined	O.T. Defined	O.T. Defined	O.T. Defined	O.T. Defined

10.4 Description of Container header contents

Words 0 to 5, Which are common to all Containers, are described in table 125.:

Sequence Number: An unsigned integer used to identify where in a series of Containers representing a video clip this Object fits.

Clip ID: An unsigned integer used for identification of a video clip. Intended for use by a higher

Table 125 – Container Header Words 0 - 5

Item	Size (Bytes)
Sequence Number	4
Clip ID	4
Container Time Stamp	8
Video Frame Rate	1
Transmission Rate	1
Reserved	2
Mode	1
Number of Objects	1
Reserved	1
Size of Extended Header	1

level protocol as a way to identify the Container as an element of a specific video clip.

Container Time Stamp: A 64 bit word used to store a FC-AV time stamp for transmission purposes. This time stamp represents the Container time. There may be other time stamps and time codes associated with each Object.

Video Frame Rate: The encoding of the Video Frame Rate field is as shown in table 126.

Transmission Rate: An 8-bit signed integer. Positive values represent n * video frame rate. Negative values represent video frame rate / n.

Mode: Simple or Extended mode.

- Bit 7 = 0, Simple mode
- Bit 7 = 1, Extended header mode

Number of Objects: The integer number of Objects in the Container.

Size of Extended Header: Valid in extended header mode only, and equal to the total number of Objects minus 4, multiplied by 16 bytes. (Positive values only).

Each Object has a 4-word field in the header to describe relevant information about that Object. The generic Object is shown in table 127.

Object n Type: See 10.6.

Table 126 – Video Frame Rate – Encoding

Frame rate per second	Code
Null	00h
15	01h
24	02h
25	03h
30	04h
30*1000/1001 (29.97 NTSC)	05h
50	06h
60	07h
60*1000/1001 (59.94 NTSC)	08h
48	09h

Table 127 – Container Header - Generic Object

Item	Size (Bytes)
Object n Type	1
Reserved	1
Object n Index	2
Object n Size	4
Object n Offset	4
Object n Object Type Defined	4

Object n Index: See 10.6.

Object n Size: Object size in bytes.

Object n Offset: Object offset in bytes from the beginning of the Container.

Object n Object Type Defined: The use of this field is defined by the Object Type.

10.5 Extended header description

The FC-AV Container system defines an extended header to address requirements of applications not directly supported by the simple mode. Provisions include the support of a larger and variable number of Objects.

The Extended Header is a logical extension of the Container header. It is physically located directly after the Container header and before Object 0. Object descriptions follow the same format as the Object descriptions within the Container header. There are no restrictions as to the Object type for any of the Objects addressed by the extended header. The length of the Extended header is dependent on the number of Objects in the Container. The extended header length equals 4 words for each Object beyond Object 3. The maximum number of Objects is 16, thus the maximum length is 4 words * (16-4)=48 words.

Table 128 – FC-AV Extended Container Header

Word	Identifier	Byte 0	Byte 1	Byte 2	Byte 3
22	Object 4 Class	Type xxh	Reserved	Index (xxh)	Index (xxh)
23	Object 4 Size	MSB	-size-	-size-	LSB
24	Object 4 Offset	MSB	-offset-	-offset-	LSB
25	Object 4 O.T. Defined	O.T. Defined	O.T. Defined	O.T. Defined	O.T. Defined
26	Object 5 Class	Type xxh	Reserved	Index (xxh)	Index (xxh)
27	Object 5 Size	MSB	-size-	-size-	LSB
28	Object 5 Offset	MSB	-offset-	-offset-	LSB
29	Object 5 O.T. Defined	O.T. Defined	O.T. Defined	O.T. Defined	O.T. Defined
	...				
n*4+6	Object n Class	Type xxh	Reserved	Index (xxh)	Index (xxh)
n*4+7	Object n Size	MSB	-size-	-size-	LSB
n*4+8	Object n Offset	MSB	-offset-	-offset-	LSB
n*4+9	Object n O.T. Defined	O.T. Defined	O.T. Defined	O.T. Defined	O.T. Defined

10.6 Object classification system

The FC-AV Container system provides a mechanism for defining the bit and byte packing of audio-video data types. It does not define audio or video formats, but rather the packing of “standard” formats into Objects for efficient transport by Fibre Channel.

Object classification is represented in the header by the Object Class Word. This word is composed of two key elements, the Type byte and the Index bytes. The Type byte is a coarse classification of the data type. Each value of Type Byte represents a table of data types. The Index Bytes key into that table.

In order to minimize duplication of effort a bit/byte packing numbering system has been developed. Entries to Type Tables indicate bit/byte packing schemes by using a value from the Packing Table.

10.6.1 Object Type byte

Object type bytes are an unsigned byte. The value represents a table. Currently assigned values are given in table 129.:

Undefined Object Type byte numbers are reserved.

10.6.2 Requirements for assignment of Object Type Byte

a) The scope of the type number is chosen to cover data types likely to be decoded by a dedicated device.

b) A definition for the function of Object Type Defined Word of the 4 word Object information block (see table 127) is required.

10.6.3 Object Index bytes

Object index bytes are defined to be a 2 byte unsigned value representing an index to the table indicated by the Object Type byte of the same Object Class Word. For the case of a null Object, the bytes are set to 0000h.

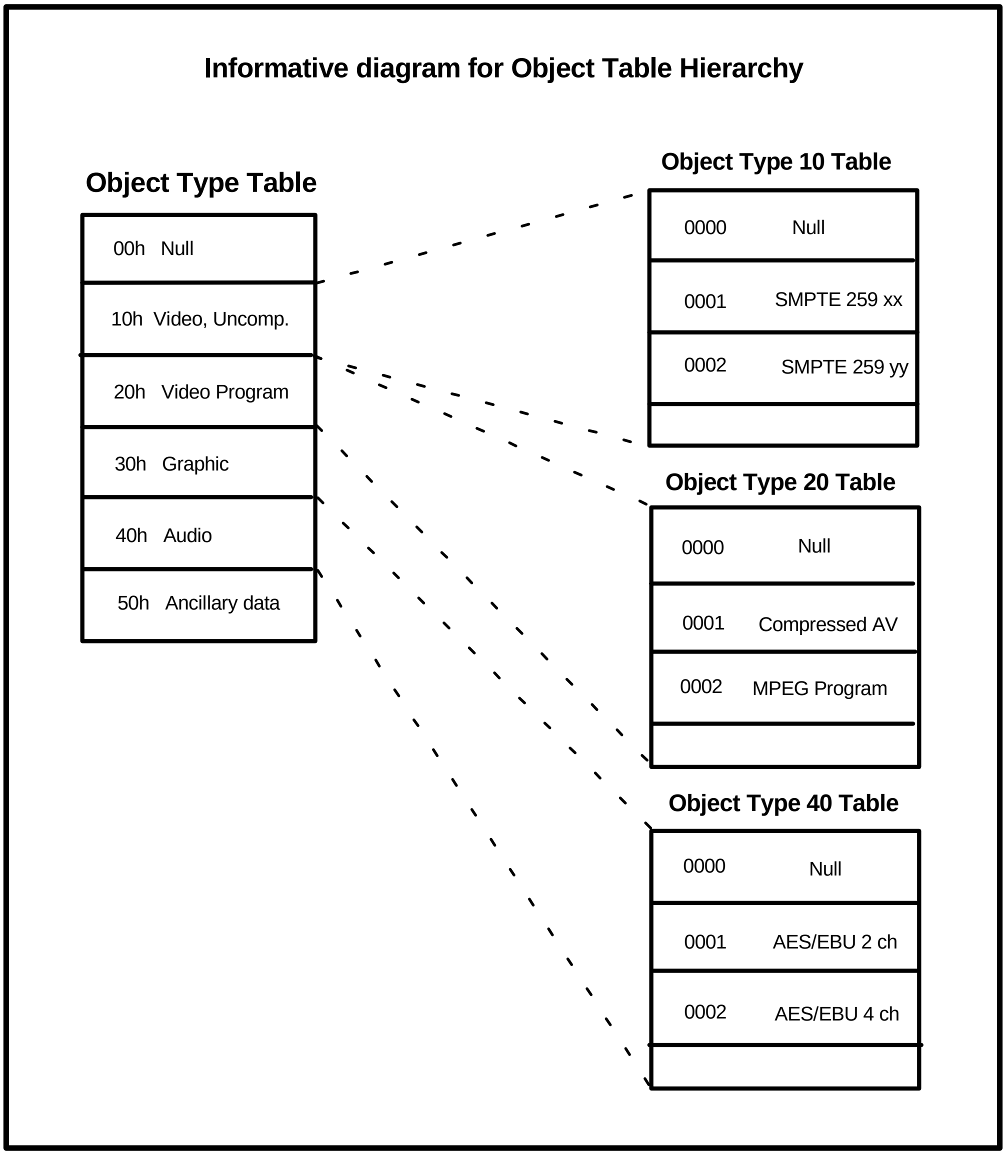

Figure 27 – Object Table Hierarchy

10.6.4 Requirements for assignment of Object Index values

a) A clear description of the data format standard to be represented by this index. Normally this would an recognized standard number. When available this may be described by SMPTE Universal Labelling System.

b) Sufficient information to describe a single data type. For example, SMPTE 259 would not be sufficient as many different types are described. A SMPTE Universal label proposed type identifier for a specific SMPTE 259 type would be sufficient providing the definition of the type is readily available

– Example: SMPTE 259M type A (270 Mbps, 720x486, 4:3, 59.97, 2:1)

c) A bit/byte packing definition for the encapsulation of this data type. References to FC-AV Packing table shall be used for this purpose.

Table 129 – Object Type – Encoding

Type	Code
Null Object	00h
Video, uncompressed	10h
Video, compressed	11h
Video Program (multiplexed stream)	20h
Graphic	30h
Audio, uncompressed	40h
Audio, compressed	41h
Ancillary data	50h

d) An indication of Object size range is desired.

10.6.5 Packing table

The packing table provides a shorthand method of describing bit and byte packing for the transmission words in the Object tables. Fibre Channel specifies a big Endian byte packing for the defined 32-bit transmission words. In general the packing and un-packing is done at driver levels and is invisible to the file system. As such fibre channel is used successfully by both big and little Endian systems. The packing definitions follow fibre channel definitions to minimize any potential system level issues.

Packing style for a particular data type (Type # & Index) is described in the Index table entry.

Table 130 – Packing Table

Description	Code
8-bit samples - Data is to be packed in Fibre Channel order (big Endian), four to a transmission word.	00h
10-bit video samples - Data is packed 3 samples to each 32-bit transmission word. See figure 28. Samples are packed in the order received, See figure 30	10h
16-bit raw sample pairs - Channel A most significant 16 bits, channel B least significant 16 bits of a 32-bit transmission word.	40h
20-bit raw audio samples - word style. See figure 29.	41h
24-bit raw audio samples - word style. See figure 29	42h
32-bit raw audio samples - word style. See figure 29	43h
AES/EBU Subframes, grouped as AES/EBU frames into 2 words. See figure 31 and figure 32	

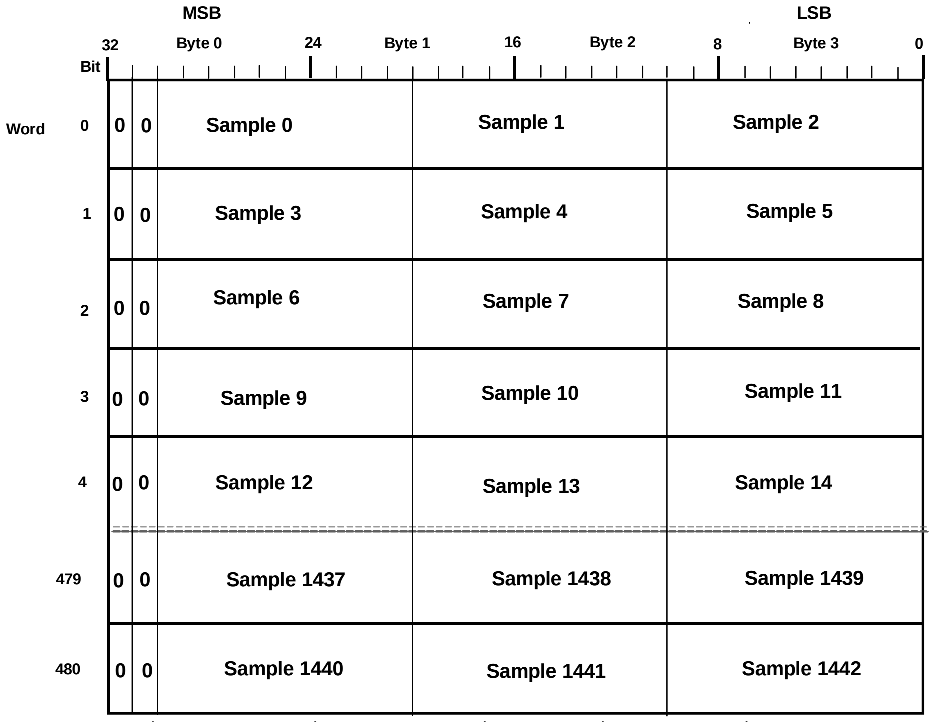
MSB
LSB
Bit
32
Byte 0
24
Byte 1
16
Byte 2
8
Byte 3
0
Word
0
0 0 Sample 0 Sample 1 Sample 2
1
0 0 Sample 3 Sample 4 Sample 5
2
0 0 Sample 6 Sample 7 Sample 8
3
0 0 Sample 9 Sample 10 Sample 11
4
0 0 Sample 12 Sample 13 Sample 14
479
0 0 Sample 1437 Sample 1438 Sample 1439
480
0 0 Sample 1440 Sample 1441 Sample 1442

Figure 28 – 10 bit packing

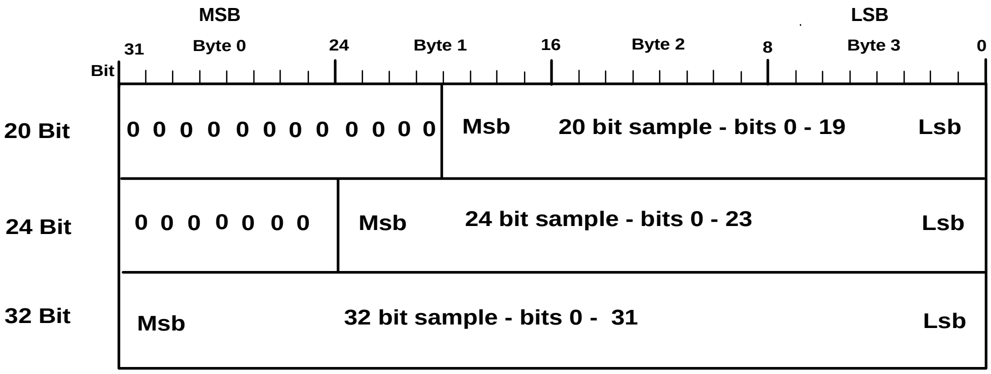

Figure 29 – Raw audio sample packing

MSB			LSB
Sample A - LSB	Sample A MSB	Sample B LSB	Sample B MSB

Figure 30 – Raw 16-bit packing of two channels, A & B

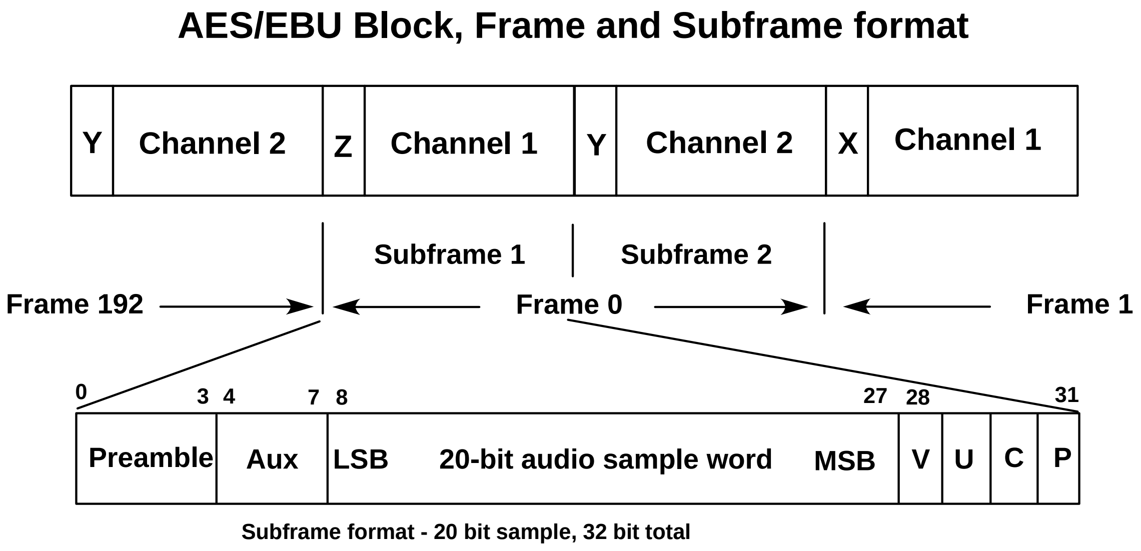

Figure 31 – AES/EBU Block, Frame and Subframe format

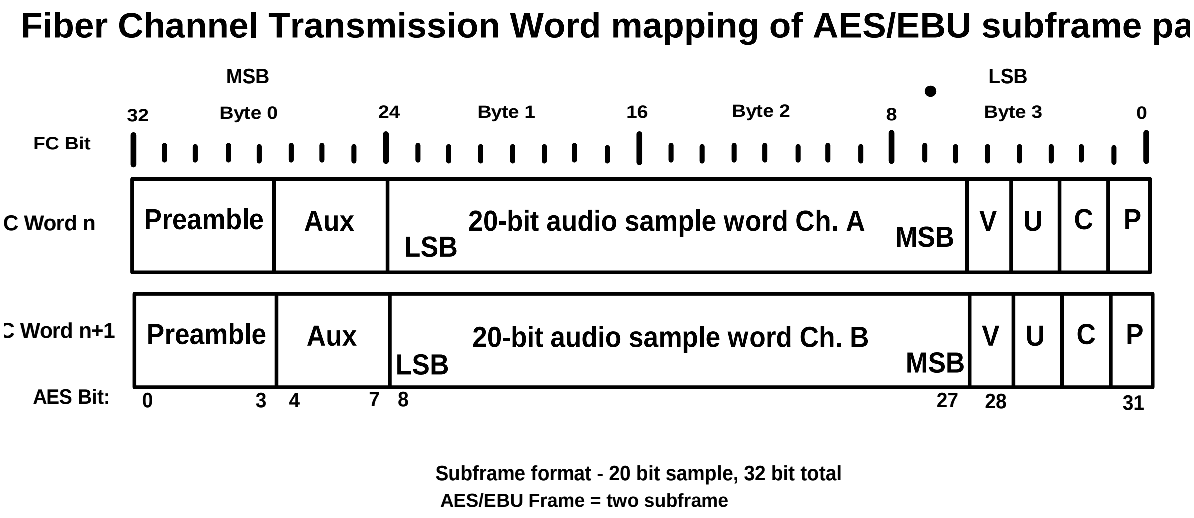
Fiber Channel Transmission Word mapping of AES/EBU subframe pa
MSB
LSB
32
Byte 0
24
Byte 1
16
Byte 2
8
Byte 3
0
FC Bit
C Word n
Preamble
Aux
LSB
20-bit audio sample word Ch. A
MSB
V
U
C
P
C Word n+1
Preamble
Aux
LSB
20-bit audio sample word Ch. B
MSB
V
U
C
P
AES Bit:
0
3
4
7
8
27
28
31
Subframe format - 20 bit sample, 32 bit total
AES/EBU Frame = two subframe

Figure 32 – Fibre Channel transmission word mapping of AES/EBU subframe pair

Annex A
(normative)

Television video primer

A.1 Television video

A.1.1 Video concepts

A.1.1.1 Fields, frames, and interlace

Television signal formats reflect the requirements of the cathode ray tube as a display device. In television systems, the scanning process consists of rapid horizontal sweeps across the cathode ray tube known as lines. The lines are spaced one below another by a slower paced vertical sweep.

In most broadcast systems, the scanning process is interlaced 2:1. In 2:1 interlace, the vertical scanning rate is doubled so that wider spaces occur between the scanned lines. The scan is stopped half way across the bottom of the image and re-inserted half way across the top of the image, and coincidentally, half way between the previously scanned lines. Systems using 2:1 interlace require half the video bandwidth of non-interlaced systems for roughly equivalent image quality.

Each complete vertical scan constitutes a field and two vertical scans constitute a frame. A frame is a single image in a video stream. By convention, field 1 begins on a full line and ends on a half line. Field 2 begins on a half line and ends on a full line.

A.1.1.2 Gamma correction

The cathode ray tube is not a linear device. Brightness increases with approximately the 2.2 power of the signal voltage. This non-linear characteristic is known as gamma. In practice, an inverse law, called gamma correction, is applied in the camera so that no non-linear circuitry is needed in the television receiver. The video stream flowing from the camera to the receiver is effectively compressed by gamma correction.

A.1.1.3 Color space

Color space refers to the mathematical representation of a color.

A.1.1.3.1 RGB

Red, green, and blue are three primary additive colors - that is, they can be added together in various proportions to form any desired color. In the RGB color space, the red, green and blue color components are each represented by an 8-bit binary value. This representation is often identified as “true color”.

A.1.1.3.2 YUV

In the YUV color space, black-and-white information (Y) is combined with color information (U and V) in such a way that a black-and-white receiver can display normal black-and-white images. Color receivers combine the additional color information with the black-and-white to produce color images.

The Y, U, and V components are derived mathematically from gamma corrected RGB. For digital RGB values in the range of 0 to 255, Y has a range of 0 to 255, U a range of -112 to 112, and V a range of -157 to 157.

A.1.1.3.3 YC_RC_B

The YC_RC_B color space was developed as part of Recommendation ITU-R BT.601. It is a scaled and offset version of the YUV color space. Y has a nominal range of 16 to 235; C_R and C_B have a range of 16 to 240 with 128 interpreted as zero.

A.1.1.4 Ancillary data

A.1.2 Video formats

There are four common video format types:

a) Composite Analog (NTSC, PAL, SECAM)

b) Component Analog (RGB, YUV)

c) Composite Digital ($4f_{sc}$)

d) Component Digital (4:2:2 or CCIR 601)

In the analog domain, composite signals are carried on a single cable and component signals are carried on three cables. In the digital domain, the composite or component analog signals are sampled at a fixed rate and transferred as a serial bit stream. Each digital sample of the analog waveform is an 8-bit or a 10-bit value.

The four common video formats are described briefly below:

A.1.2.1 Composite analog video

At the time of the creation of color television systems, it was essential that the color system be fully compatible with the existing black-and-white system. To accomplish this, techniques evolved in which the picture and brightness information (luminance) was given a broadband representation and the color information (chrominance) was given a relatively narrow-band representation.

Because the color component does not carry a perfect representation of all the color information the camera sees, various techniques evolved to encode the color information and to combine that information with the picture and brightness information. The common analog techniques for this are known as NTSC, PAL, and SECAM.

NTSC stands for National Television System Committee, a committee of the Electronic Industries Association (EIA). Its work was endorsed by the Federal Communications Commission (FCC) in 1953 for commercial broadcasts in the United States. Subsequently, NTSC was adopted by Canada, Japan, Mexico, the Philippines, and other Central American and Caribbean area countries.

In 1967, PAL, which stands for Phase Alternation Line was adopted in Europe and the United Kingdom. Also In 1967 SECAM, which stands for Sequentiel Couleur Avec Memoire, was adopted by France and the then USSR.

The line and field rates associated with these three encoding schemes may vary considerably with local monochrome standards. In this document, however, two broadly accepted formats are of interest:

a) NTSC 525
   - 525 lines per video frame
   - 59.94 fields per second

b) PAL 625
   - 625 lines per video frame
   - 50 fields per second

A.1.2.2 Component analog video

A.1.2.3 Composite digital video

In composite digital video systems, the analog waveform is sampled at four times the color subcarrier frequency, hence the designation $4f_{sc}$.

The color subcarrier frequency in NTSC is 3.5795454 MHz; the 4x sampling rate for NTSC is approximately 14.32 MHz. 10-bit samples, taken at this sampling rate produce a serial data stream of 143 Mbits/sec.

The color subcarrier frequency in PAL is 4.43361875 MHz; the 4x sampling rate for PAL is approximately 17.73 MHz. 10-bit samples, taken at this sampling rate produce a serial data stream of 177 Mbits/sec.

ANSI/SMPTE Standard 259M describes a 10-bit encoding of the $4f_{SC}$ composite signals.

A.1.2.4 Component digital video

ITU-R recommendation BT.601-4, formerly and commonly known as CCIR 601, is an International Telecommunication Union, Radiocommunications Sector (ITU-R) standard for component digital video widely deployed in the television industry.

In BT.601-4, the component signals are identified as luminance (Y) and two color difference signals (C_R, C_B). The sampling frequency for the luminance signal is 13.5 MHz and the sampling frequency for each color difference signal is 6.75 MHz. When the three signals are multiplexed, the resulting sampling frequency is 27 MHz.

BT.601-4 also describes the rules for constructing Y, C_R, and C_B from the analog signals. The form of coding is a uniformly quantized PCM, 10 (formerly 8) bits per sample.

In the jargon of component digital video, the ratio 4:2:2 is often used to describe this video format. The concept underlying this notation is that the luminance sampling rate is 4x an imaginary subcarrier frequency and each of the color difference sampling rates are 2x an imaginary subcarrier frequency - hence 4:2:2.

ANSI/SMPTE Standard 259M, often called Serial Digital Interface (SDI), describes a 10-bit encoding of the 4:2:2 component signals. The bit rate required for the resulting serial data stream is 270 Mbits/sec.

ANSI/SMPTE Standard 125M describes a multiplexing format for the 10-bit samples derived from ANSI/SMPTE Standard 259M. It also describes timing reference signals called end-of-active-video (EAV) and start-of-active-video (SAV), which are inserted in the data stream as markers.

Ancillary data is often placed in vendor unique ways between the EAV and SAV markers. Ancillary data may include digital audio channels, time code, and error detection and handling information.