The ATM Forum

Technical Committee

Frame Based ATM over SONET/SDH Transport (FAST)

fb-fbatm-0151.000

July, 2000

The ATM Forum
Worldwide Headquarters
2570 West El Camino Real, Suite 304
Mountain View, CA 94040-1313
Tel: +1-650-949-6700
Fax: +1-650-949-6705

Acknowledgments

The SAA/FUNI working group was chaired by initially by Doug O'Leary, then by Andy Malis. It eventually became a working group in its own right ("Frame Based ATM" or FBATM), chaired by Andy Malis. Tim Dwight was editor for the *Frame-Based ATM over SONET/SDH* specification.

The following people made significant technical contributions to the *Frame-Based ATM over SONET/SDH* specification:

Matthew Bocci
Arnold Bragg
John Drake
Tim Dwight
Juha Heinänen
Anthony Hung
John Kenney
Andy Malis
Brian McBride
Bruce Northcote
Doug O'Leary
Hossain Pezeshki-Esfahani
Kenneth J. Rehbehn
David Wells

Table of Contents

List of Figures

1 Introduction

This document defines the mechanisms and procedures required to support the transport of variable length datagrams, hereafter known as *ATM frames*, over an ATM infrastructure utilizing high-speed SONET or SDH transmission facilities. Existing ATM control and management plane functions (e.g., signaling, routing, addressing, ILMI, OAM) are supported, ensuring interoperability with and leveraging the power of, capabilities developed for use over cell-based interfaces.

Both UNI and NNI interfaces are specified. The term *FAST* is used to refer to an interface compliant with this specification, while *FAST UNI* and *FAST NNI* are used to differentiate UNI and NNI interfaces.

FAST interfaces are functionally equivalent to their cell-based counterparts, with the following exception:

– The maximum SDU size for which support is required at a FAST UNI is 9216 octets. Maximum SDU size on a cell-based UNI is negotiated between end systems, up to the bounds set by the adaptation layer.

Two encapsulation formats are defined for use over FAST interfaces. The first, known as *frame encapsulation*, involves the transmission of variable-length frames. The second, known as *cell encapsulation*, encapsulates the payload of a single ATM cell within the standard FAST framing. A FAST NNI must support both formats. A FAST UNI may support the transport of user data using either encapsulation, but is only required to support transport of user data via frame encapsulation. OAM information is transported using cell encapsulation.

Two modes of operation are defined. The first, known as *mode 0*, is both conceptually simpler and more efficient in terms of bandwidth use, than *mode 1*. Mode 1 is more generally applicable, however. Details of the capabilities of the two operational modes, are presented in section 3.

In the following text **(R)** and **Shall** indicate a requirement that must be implemented to meet this specification. **(O)** and **May** indicate an Option that may be implemented. **(CR)** indicates a Conditional Requirement which must be implemented if the option to which it is related, is implemented.

1.1 Relationship to the Frame-based User-to-Network Interface (FUNI)

The ATM Forum has authored several frame-oriented specifications. The first was af-saa-0014.000 [DXI] which defined the ATM Data eXchange Interface (ATM DXI). This document described a DTE-to-DCE interface modeled after the SMDS DXI. Specification af-saa-0031.000 [FUNIv1] defined a simplified subset of ATM DXI. Specification af-saa-0088.000 [FUNIv2] superseded af-saa-0031.000, providing clarification and extending capabilities in ways which were neither precluded nor explicitly allowed in af-saa-0031.000. This specification was subsequently extended by af-saa-0109 [FUNI.mse], which defined mechanisms by which real-time and multimedia applications (e.g., voice) may be supported over a FUNI interface.

Frame-Based ATM over SONET/SDH is similar in many respects to these specifications. Its two major departures are:

1. While DXI and FUNI interfaces are designed to operate over low-speed plesiochronous transmission facilities, Frame-Based ATM over SONET/SDH is designed to operate over high-speed SONET/SDH facilities.
2. DXI and FUNI interfaces provide access to an ATM network. Frame-Based ATM over SONET/SDH interfaces may be used to provide network access, inter-switch trunking or both.

Table 1 summarizes the differences between Frame Based ATM over SONET/SDH and the FUNI specification (af-saa-0088.000). The capabilities of a cell-based interface are also illustrated for comparison.

		FAST		
	[cell-based] ATM	Frame encapsulation	Cell encapsulation	FUNI
Physical Layers	Various	SONET/SDH	SONET/SDH	DS1/E1, DS3/E3
Adaptation Layers	All	AAL5	All	AAL5, AAL3/4
ATM Layer Service Categories	All	All	All	NRT-VBR, UBR
Usage Parameter Control	Defined			N/A[1]
Connection types	VCC, VPC	VCC	VCC, VPC	VCC
Interworking with cell-based interfaces	N/A	Partial (mode 0)[2] Full (mode 1)	Full	Partial[3]
OAM support	Full	Partial (mode 0)[4] Full (mode 1)	Full	Partial[4]
Maximum SDU size (bytes)	≤ 65535, depending on adaptation layer	*9216 £max £65535*[6]	≤ 65535, depending on adaptation layer	4096 (mode 1) 9232 (modes 3,4)
CRC size (bytes)[5]	1	4	4	2 (mode 1) 4 (modes 3,4)

Table 1 - Comparison of FAST and FUNI

Notes:

(1) FUNI mandates that the IWF shape the transmit rate in the frame-to-cell direction, but neither defines nor requires the implementation of a UPC function with respect to the frame traffic inbound to the IWF.

(2) FAST mode 0 does not support the CPCS Common Part Indicator (CPI) octet, which is intended to enable communication between AAL5 CPCS entities.

(3) FUNI does not support the CPCS User-to-User (UU) octet defined in I.363.5 for the transparent conveyance of information between users of the AAL5 CPCS.

(4) Neither FUNI nor FAST mode 0 can transparently support OAM applications which insert OAM cells within the payload of the CPCS_PDU. In either case, such cells will be transmitted by the IWF towards the frame-based interface, in advance of the frame in whose midst they arrived.

(5) References to CRC above are with respect to the link layer (i.e., HEC for cell-based ATM, HDLC FCS for FUNI and FAST).

(6) In some implementations the HDLC layer may impose constraints which limit the SDU size supportable via frame encapsulation. To be compliant with this specification, however, an SDU of 9216 must be realizable.

1.1.1 Interoperability

The functionality supported at a FAST UNI is a superset of that supported at a FUNI. Connections whose endpoints are a mixture of FAST UNIs and FUNIs can support exactly the same applications as can be supported by a connection all of whose endpoints are FUNIs.

Connections originating on a FUNI and cross-connected on the network side of the FUNI to a FAST NNI, require the “back to back IWF” configuration illustrated in Figure 1. The FUNI IWF is responsible for shaping the traffic transmitted into the FAST IWF, thereby ensuring conformance with the connection’s traffic parameters and enabling QoS to be guaranteed downstream. This is described in section 9.2.2 of [FUNIv2]. The two interworking functions illustrated in Figure 1 may be within a single network element, or separated by an arbitrary number of cell-based ATM interfaces.

The FAST IWF is described in section 2.

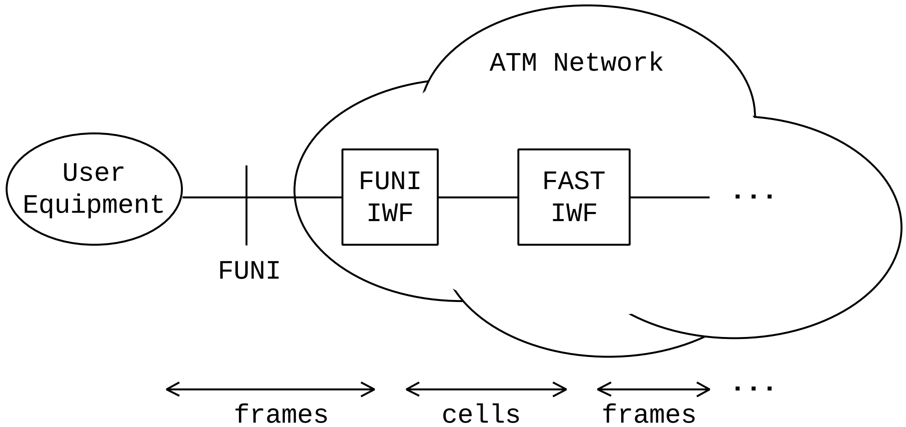

Figure 1 - FUNI to FAST Interworking

2 Reference Model

Figure 2 illustrates the functional reference model for Frame Based ATM over SONET/SDH, as defined in this specification.

The FAST UNI and FAST NNI interfaces defined herein can be used end to end, given a network so configured. Upper layers may access the FAST Data Link layer directly or through a modified AAL5 Common Part Convergence Sublayer (CPCS). The interface presented by the AAL5-CPCS to the upper layers is as specified in ITU-T recommendation I.363.5, however the downward interface is modified to accommodate the FAST Data Link Layer rather than the AAL5 SAR sublayer. Inter-layer interfaces are discussed in section 5.

Communication between CPCS entities (via the AAL5 Common Part Indicator) is supported by mode 1 but not by mode 0.

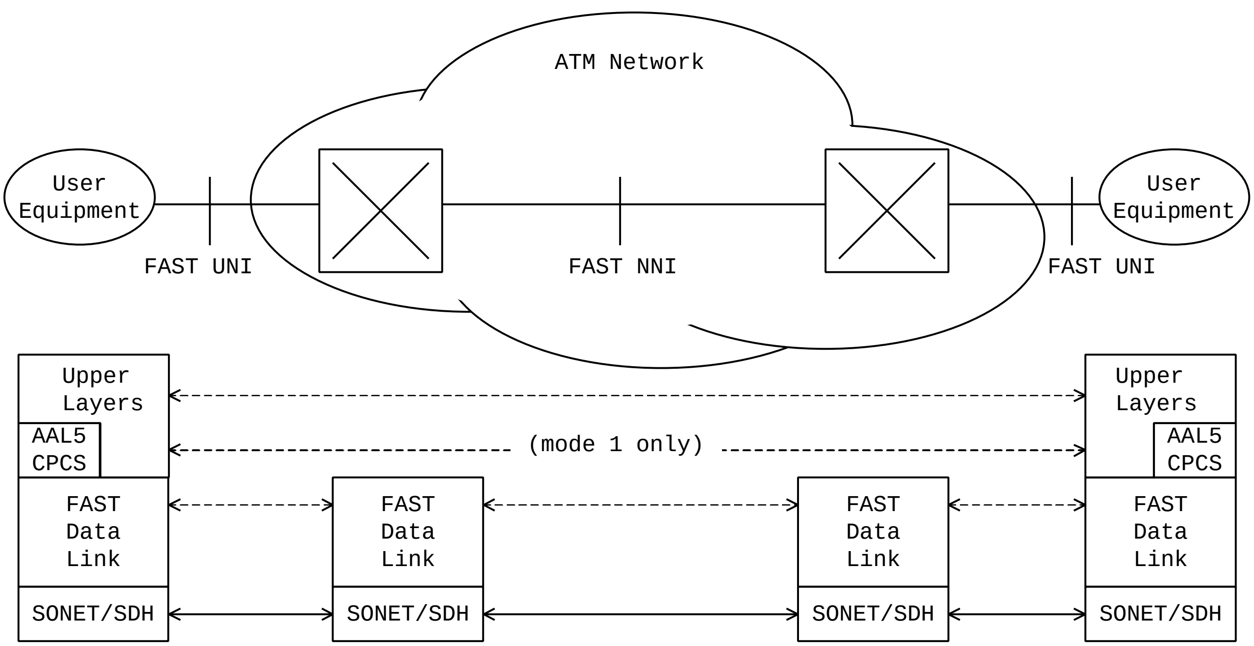

Figure 2 - FAST Reference Model

In order to support the establishment of ATM connections across a mixture of FAST and cell-based (or FUNI-based) interfaces, a FAST interworking function (IWF) is required. The FAST IWF maps between the data link layers in a manner designed to preserve ATM layer semantics (e.g., CLP and EFCI markings) and may depending on the attributes of the connection and the configuration of the IWF, modify the encapsulation of the data from cells to frames or vice versa.

The FAST IWF may be deployed in three locations:

- At the network edge, joining a cell-based UNI to a FAST NNI (as noted in section 1.2.1 and illustrated in Figure 1, FUNI to FAST NNI interworking is a special case of this configuration).
- Within the network, joining a FAST NNI to a cell-based NNI.
- At the network edge, joining a FAST UNI to a cell-based NNI.

The options for deployment of the FAST IWF are illustrated in Figure 3.

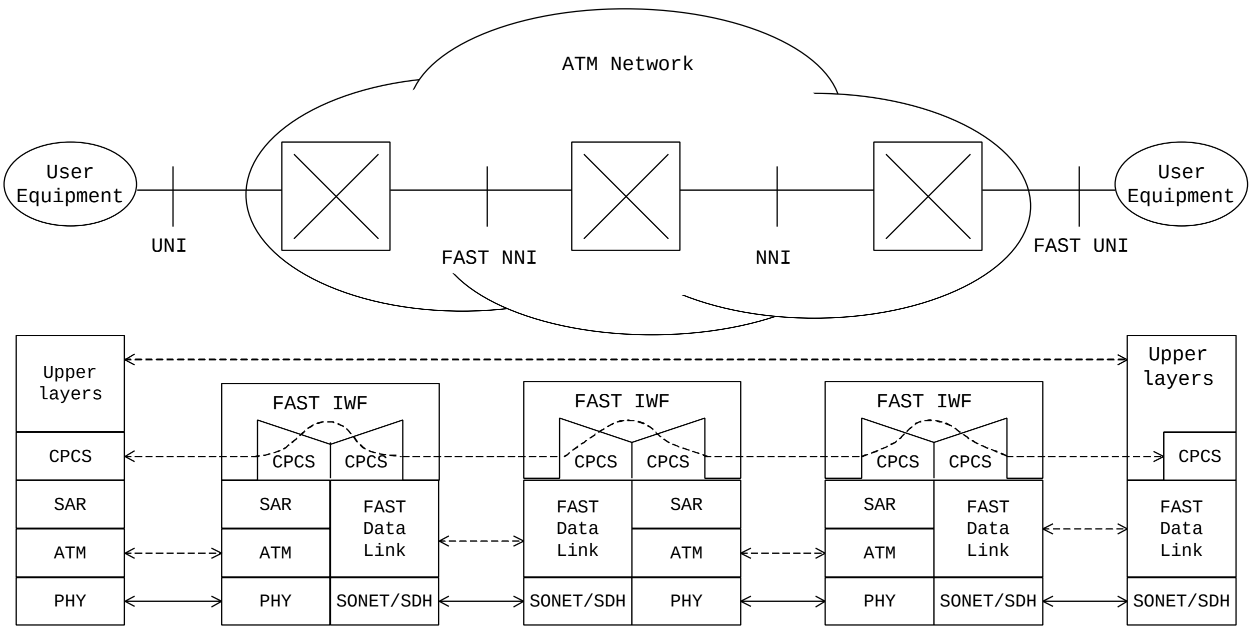

Figure 3 - FAST IWF Deployment Options

3 Data Link Layer Protocol

3.1 Frame Format

This specification uses a simplified version of the frame structure defined in RFC-1662, “PPP in HDLC-like Framing”, omitting the address, control, protocol and padding fields defined therein. It also mandates the use of a 32 bit frame check sequence (FCS), which is optional in RFC-1662. The result is shown in Figure 4.

flag	information	FCS	flag
01111110	variable length	32 bits	01111110

Figure 4 - FAST frame format

The *information* field is structured differently depending on whether the associated interface utilizes mode 0 or mode 1. The formats are defined in sections 3.1.1 and 3.1.2.

(R1) A FAST implementation *shall* utilize the frame format illustrated in Figure 4.

(R2) A FAST implementation *shall* support either mode 0 or mode 1. It *may* support both.

3.1.1 Coding of the Information field (Mode 0)

The information field of the mode 0 frame structure is illustrated in Figure 5. This structure is used for both frame and cell encapsulation, when operating in mode 0.

	8	7	6	5	4	3	2	1
1	GFC (UNI) or VPI (NNI)				VPI			
2	VPI				VCI			
3	VCI							
4	VCI				PTI			CLP
5	User data or OAM cell payload							
.								
.								
.								
.								
.								
.								
.								
n	User to User Octet							

Figure 5 – format of the *information* field (mode 0)

Note 1: $5 \le n \le (max\_SDU\_size + 5)$

The first 4 octets are identical to the first 4 octets of an ATM cell. Except where explicitly noted in this specification, requirements associated with their use are identical to those applicable to ATM cells.

The User to User Octet may be utilized by the application (i.e., the user of the adaptation layer) to transport information between connection end points. When interworking between frame-based and cell-based interfaces, this octet is mapped to the AAL5 CPCS_UU octet.

When using *cell encapsulation* the User to User Octet is not used. Its value must be set to zero by the transmitter, and must not be verified by the receiver.

(CR3) If the implementation supports mode 0, it *shall* code the *information* field of the FAST frame as described in section 3.1.1 above, and illustrated in Figure 5.

3.1.2 Coding of the Information field (Mode 1)

The information field of the mode 1 structure is illustrated in Figure 6. This structure is used for both frame and cell encapsulation, when operating in mode 1.

<table>
<tr><th></th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th></tr>
<tr><td>1</td><td colspan="4">GFC (UNI) or VPI (NNI)</td><td colspan="4">VPI</td></tr>
<tr><td>2</td><td colspan="4">VPI</td><td colspan="4">VCI</td></tr>
<tr><td>3</td><td colspan="8">VCI</td></tr>
<tr><td>4</td><td colspan="4">VCI</td><td colspan="3">PTI</td><td>CLP</td></tr>
<tr><td>5</td><td>B</td><td>E</td><td>C</td><td>R</td><td colspan="4">Sequence No. (high 4 bits)</td></tr>
<tr><td>6</td><td colspan="8">Sequence No. (low 8 bits)</td></tr>
<tr><td>7
8</td><td colspan="8">Cell Position Indicator (CPI)</td></tr>
<tr><td>9
.
.
.
.
.
.
n-m</td><td colspan="8">User data or OAM cell payload</td></tr>
<tr><td>n-m+1
n-m+2
.
.
.
n</td><td colspan="8">Trailer</td></tr>
</table>

Figure 6 – format of the *information* field (mode 1)

Note 1: $56 \le n \le (8 + (\lceil (max\_SDU\_size + 8) \div 48 \rceil \times 48))$
where:
$\lceil x \rceil$ refers to the smallest integer greater than or equal to x

Note 2: $8 \le m \le 55$

The first 4 octets are identical to the first 4 octets of an ATM cell. Except where explicitly noted in this specification, requirements associated with their use are identical to those applicable to ATM cells.

Octets 5 and 6 are used when fragmenting and reassembling large frames. They jointly comprise the mode 1 *fragmentation header*, which is discussed in section 3.8.

The Cell Position Indicator (CPI) field is only meaningful when the payload is an OAM cell (PTI=1xx). In that case, it enables the frame-to-cell interworking function to place OAM cells within a sequence of user data cells, at the proper relative position. This process is defined in section 8.

The mode 1 payload and trailer jointly comprise an AAL5 PDU (i.e., the trailer is identical to that of an AAL5 CPCS_PDU). The trailer is formatted in accordance with ITU-T Recommendation I.363.5.

When a connection is transported using *cell encapsulation*, the Trailer field is not present. Octets 5-6 are coded as specified in section 3.8. Octets 7-8 may be non-zero only in OAM cells associated with a connection transported using *frame encapsulation.*

(CR4) If the implementation supports mode 1, it *shall* code the FAST frame as described in section 3.1.2 above, and illustrated in Figure 6.

3.1.3 Header Processing

Unless otherwise indicated by this specification, the requirements applicable to the ATM layer of a cell-based interface, are applicable to a FAST interface.

(R5) The GFC field *shall* be encoded as 0000 prior to transmission over a FAST UNI. Its value shall be ignored upon receipt.

(R6) Virtual Path (VP) and Virtual Channel (VC) requirements for FAST interfaces, *shall* be identical to those for the corresponding (UNI or NNI) cell-based interfaces.

3.1.3.1 Requirements at the FAST IWF

The following requirements are applicable when a connection is cross-connected between a cell-based and a FAST interface.

Such connections may be transported over the FAST interface using either *frame encapsulation* or *cell encapsulation*. In the case of *cell encapsulation*, the FAST header and the first four bytes of the corresponding cell header, are identical[1]. When *frame encapsulation* is used, the IWF must apply the rules specified below to map information between the respective headers.

Within connections transported using frame encapsulation, there may be cells (e.g., OAM cells) which cannot be reassembled at the IWF. Header translation for such cells is identical to that performed for connections utilizing cell encapsulation.

(R7) For cells which are not reassembled by the IWF (either because they belong to a connection transported using cell encapsulation, or because their PTI value of 1xx indicates them not to be user data cells), or frames which are not segmented by the IWF (for the same reasons) the FAST IWF *shall* preserve the first four bytes of the header.

The following requirements apply to the processing of cells which are reassembled, or frames which are segmented, by the IWF.

(R8) The value of the left-most PTI bit *shall* be preserved (i.e., the value leaving the FAST IWF shall equal the value entering the FAST IWF).

(R9) The Explicit Forward Congestion Indication (EFCI, center bit of the PTI field) *shall* be processed as follows:

Cell-to-frame direction	Following AAL5 reassembly, where the leftmost bit of the PTI is zero, EFCI shall be set to one in the resulting frame if local congestion is to be indicated, or if the EFCI indication was set to one in the last constituent cell. Otherwise EFCI shall be set to zero.
Frame-to-cell direction	Following AAL5 segmentation, the EFCI indication in the resulting cells shall be set to the value to which EFCI was set in the corresponding frame.

[1] This requirement does not preclude VPI/VCI translation, EFCI marking, CLP marking, or any other function which may legitimately alter values found in the header. It merely indicates that such functions are implemented either before or after traversing the IWF.

(R10) The SDU_type indication (right-most bit of the PTI field) *shall* be processed as follows:

Cell-to-frame direction	Following AAL5 reassembly, the SDU_type indication in the resulting frame *shall* be set to 1.
Frame-to-cell direction	Following AAL5 segmentation, the SDU_type indication *shall* be set in accordance with the AAL5 specification (i.e., set to one in the last cell of the frame, and to zero in all other cells).

(R11) The Cell Loss Priority (CLP) bit, *shall* be processed as follows:

Cell-to-frame direction	Following AAL5 reassembly, CLP *shall* be set to one in the resulting frame if CLP was set to one in any of the constituent cells. Otherwise CLP shall be set to zero.
Frame-to-cell direction	Following AAL5 segmentation, CLP in the resulting cells *shall* be set to the value to which CLP was set in the corresponding frame.

3.1.4 Trailer Processing

Mode 0 and Mode 1 differ with respect to the structure and processing of the FAST trailer.

3.1.4.1 Mode 0 Requirements

The mode 0 frame trailer format (Figure 5, page 11) contains a single one-byte field. Its purpose is to carry information transparently between connection endpoints, external to the SDU. Interlayer procedures through which higher layers gain access to this field, are discussed in section 5.

(R12) A FAST NNI *shall not* modify the value of the User to User octet.

At a FAST IWF, usage of the frame trailer differs according to the encapsulation mechanism in use. In the case of cell encapsulation, the trailer is not used. Its value must be set to zero by the transmitter and must not be verified by the receiver. In the case of frame encapsulation, the User-to-User Indication (CPCS_UU) field of the AAL5 CPCS_PDU is mapped to the User to User octet of the mode 0 FAST trailer. Remaining fields of the AAL5 CPCS_PDU (*Padding*, *Common Part Indicator*, *Length* and *CRC*) are discarded in the cell-to-frame direction and generated anew in the frame-to-cell direction.

(R13) For cells which are not reassembled by the FAST IWF (either because they belong to a connection transported using cell encapsulation, or because their PTI value of 1xx indicates them not to be user data cells), the value of the User to User octet in the resulting FAST frame *shall* be set to zero by the transmitter and *shall not* be verified by the receiver.

(R14) For cells which it reassembles, the FAST IWF *shall* map the User to User octet from the FAST trailer to the CPCS_UU field of the AAL5 CPCS_PDU, as follows.

Cell-to-frame direction	The User to User Data value obtained from the AAL5-CPCS UNITDATA.signal primitive *shall* be transferred to the User to User octet of the FAST trailer.
Frame-to-cell direction	The User to User octet value obtained from the FAST trailer *shall* be passed to the AAL5 CPCS as parameter *User to User Data* in the AAL5-CPCS UNITDATA.invoke primitive.

3.1.4.2 Mode 1 Requirements

The mode 1 frame format (Figure 6 page 12) defines a trailer which is structurally and semantically identical to that of AAL5. Associated requirements are defined in ITU-T recommendation I.363.5. Interlayer procedures through which higher layers gain access to this field, are discussed in section 5.

Mode 1 preserves the value of the trailer end to end, across both NNI and FAST NNI interfaces and across FAST IWFs. To do so, it utilizes direct interfaces to the Segmentation and Reassembly (SAR) sublayer and the FAST Data Link layer (in each case, bypassing the AAL5 CPCS sublayer). Via these interfaces it extracts the AAL5 trailer from the incoming FAST frame (in the frame-to-cell direction) or AAL5 CPCS_PDU (in the cell-to-frame direction); and places it in the outgoing FAST frame (in the cell-to-frame direction) or AAL5 CPCS_PDU (in the frame-to-cell direction).

As indicated in section 3.1.2, the mode 1 trailer is not present when using cell encapsulation.

(R15) A FAST NNI *shall not* modify the value of the frame trailer.

(R16) For cells which it reassembles, the FB-ATM IWF *shall* map between the FB-ATM trailer and the AAL5 trailer contained within the CPCS_PDU, as follows:

Cell-to-frame direction	Following AAL5 reassembly, the trailer contained within the CPCS_PDU *shall* be copied to the FAST trailer.
Frame-to-cell direction	The FAST trailer *shall* be copied to the CPCS_PDU trailer, after generation of the CPCS_PDU and prior to segmentation.

3.2 Adaptation Layers

Higher layers access the FAST Data Link layer either directly or through the AAL5 CPCS. At a FAST UNI, this is the only adaptation layer whose support is required or for which supporting procedures are defined in this specification. A FAST NNI must support all connections, regardless of their adaptation layer.

Adaptation layers are not implemented at a FAST NNI, except in the case of interworking. The FAST IWF must support AAL5. Since the FAST UNI only supports AAL5, connections using adaptation layers other than AAL5 are understood to originate and terminate on cell-based interfaces. Adaptation layer information associated with such connections, is carried transparently through the IWF (using *cell encapsulation*).

(R17) A FAST UNI *shall* support the AAL5 CPCS.

(R18) Except in the case of interworking, a connection's adaptation layer *shall* be transparent at the FAST NNI.

(R19) A FAST IWF *shall* support AAL5 on the cell interface and AAL5 CPCS on the FAST interface.

(R20) Connections associated with adaptation layers other than AAL5 *shall* be transported using *cell encapsulation.*

3.3 Connection Types

All FAST interfaces support Virtual Channel Connections (VCCs). Support for Virtual Path connections (VPCs) is optional at the FAST NNI. Since the FAST UNI only supports VCCs, Virtual Path connections at a FAST NNI are understood to originate and terminate on cell-based interfaces. Such connections cannot be reassembled at the IWF, hence are transported over the FAST NNI using *cell encapsulation*.

(R21) All FAST interfaces *shall* support Virtual Channel Connections (VCCs).

(O22) A FAST NNI *may* support Virtual Path Connections (VPCs).

(CR23) If Virtual Path Connections are supported at a FAST NNI, they *shall* be transported using *cell encapsulation.*

3.4 ATM Layer Service Categories

All ATM Layer Service Categories (e.g., CBR, rt-VBR, nrt-VBR, ABR, UBR, GFR) defined in the ATM Forum's Traffic Management version 4.1 specification, may be supported at FAST interfaces. As is the case with cell based interfaces, the definition of which service categories are supported at a given interface may be subject to configuration and to the capabilities of the device.

(R24) A FAST interface *shall* support some subset of the ATM Layer Service Categories defined in the ATM Forum's Traffic Management 4.1 specification.

3.5 Encapsulation Types

A FAST interface may support AAL5 VCCs through the use of either *frame encapsulation* or *cell encapsulation*, depending on its configuration. It must at minimum support the ability to carry such connections via *frame encapsulation.* Configuration options (e.g., the selection of encapsulation format for AAL5 VCCs based on ATM layer service category) are not subject to standardization.

(R25) A FAST interface *shall* support the transport of AAL5 VCCs using *frame encapsulation*.

(O26) A FAST interface *may* support the transport of AAL5 VCCs using *cell encapsulation*.

(CR27) If a FAST interface supports transport of AAL5 VCCs via either *cell encapsulation* or *frame encapsulation*, it *shall* make the selection of encapsulation format configurable by the network operator. The default *shall* be to carry such connections using *frame encapsulation.*

Virtual Path Connections and non-AAL5 Virtual Channel Connections, may be supported at FAST NNIs. Neither is supported at the FAST UNI. Both are transported via cell encapsulation, as indicated in sections 3.2 and 3.3.

3.6 Transport of OAM Information

OAM cells are supported using cell encapsulation. Transport of OAM cells is required at the FAST NNI and optional at the FAST UNI. Details related to the support of OAM cells over FAST interfaces are provided in section 8.

(R28) A FAST NNI *shall* support the transport of OAM cells.

(O29) A FAST UNI *may* support the transport of OAM cells.

(CR30) If a FAST interface supports the transport of OAM cells, it *shall* do so using cell encapsulation.

3.7 User Service Data Unit size

(R31) A FAST interface *shall* support a maximum SDU size of at least 9216 and at most 65535 octets[2].

---

[2] This value is chosen based on the default MTU size of 9188 octets identified in the IETF's RFC 1577 (Classical IP over ATM). 9216 is the smallest multiple of 48 which is greater than or equal to 9188.

(CR32) A FAST interface which supports a maximum SDU size less than 65535 octets, *shall* process SDUs whose size exceeds the maximum SDU size, as follows:

Mode 0	SDUs whose length exceeds the interface's maximum SDU size, *shall* be discarded.
Mode 1	SDUs whose length exceeds the interface's maximum SDU size, *shall* either be discarded or fragmented into units of length less than or equal to the interface's maximum SDU size, according to the procedures defined in section 3.8 of this specification.

3.8 Fragmentation

This section describes an optional process whereby a FAST mode 1 implementation may sub-divide frames whose length exceeds a local maximum SDU size into a series of *fragments*, and reassemble a sequence of such fragments to reconstitute the original frame.

(CR33) If a FAST implementation supports fragmentation, it *shall* do so in accordance with the requirements identified in the following sub-sections.

3.8.1 Overview

To properly support voice and other real-time (delay-sensitive) data on frame-based ATM links, it may be necessary to fragment very long frames that share the same link into shorter frames. The shorter frames eliminate excessive frame delays which could be encountered with longer frames. Fragmentation enables interleaving delay-sensitive traffic on one VC with fragments of a long frame on another VC utilizing the same link.

Fragmentation and reassembly are strictly local to the link, and the fragment size can be optimally configured to provide the proper delay and delay variation based upon the link speed.

When fragmentation and reassembly are used on a link, all Mode 1 frames on all VCs (including all PVCs, SVCs, and signaling channels) must utilize the fragmentation header. If fragmentation and reassembly are not used on a link, then the fragmentation header is ignored in all Mode 1 frames on that link.

3.8.2 Fragmentation Header

Octets 5 and 6 of the mode 1 frame header (Figure 6) comprise a fragmentation header, structured as follows:

	8	7	6	5	4	3	2	1
5	B	E	C	R	Sequence No. (high 4 bits)			
6	Sequence No. (low 8 bits)							

Figure 7 – Fragmentation Header

The (B)eginning fragment bit is a one-bit field set to 1 on the first frame fragment derived from the original frame and set to 0 for all other fragments from the same frame.

The (E)nding fragment bit is a one-bit field set to 1 on the last fragment of a frame and set to 0 for all other fragments of the same frame. A frame fragment may have both the (B)eginning and (E)nding fragment bits set to 1 when the whole frame fits in one fragment.

The (C)ontrol bit is set to 0 in all fragments. It is reserved for future control functions.

The (R)eserved bit is set to 0 in all fragments. It is reserved for future functions.

The sequence number is a 12-bit binary number that is incremented modulo 2^{12} for every frame fragment transmitted on a VC. There is a separate sequence number maintained for each VC across the interface.

Peer entities on a link shall be identically configured for fragmentation functionality, i.e., both shall be configured with fragmentation enabled or both with fragmentation disabled.

3.8.3 Fragmentation Procedure

This fragmentation procedure is based on RFC 1990 [RFC1990].

A transmitter shall perform frame fragmentation according to the following procedure:

A series of frame fragments is created by removing the leading flag, the four-octet frame header, the FCS and the trailing flag from the frame and sending the remaining octets (CPI, payload, and trailer), in their original order, as a series of frame fragments.

The resulting fragments shall be transmitted in the same order as they occurred in the original frame prior to being fragmented.

Fragments from multiple VCs may be interleaved with each other on one link (this is the principal objective of fragmentation).

Every fragment in the series shall contain a fragmentation header and a copy of the frame header that was on the original unfragmented frame.

The B bit shall be set to 1 in the first fragment of each frame. The E bit shall be set to 1 in the last fragment of each frame.

Following a VC becoming active, the first fragment sent on that VC may have the sequence number set to any value (including zero). The sequence number shall subsequently be incremented by one for each fragment sent. The sequence number shall be incremented without regard to the original frame boundaries, i.e., if the last fragment in one frame used sequence number "N", then the first fragment of the following frame shall use sequence number "N+1" modulo 2^{12}. Each VC shall have its own fragmentation sequence number, independent of all other VCs. This enables detection of lost fragments (and bursts of lost fragments) independently for all active VCs.

Note that if sufficient fragments are sent on an active VC, the sequence number will wrap from all ones to zero, and will eventually also wrap past the original sequence number sent on that VC after it became active. This wrapping may or may not occur on an original frame boundary (it is transparent to frame boundaries).

3.8.4 Reassembly Procedure

A fragmentation receiver shall perform reassembly of received frame fragments into the original frames according to the following procedure:

For each VC, the receiver shall keep track of the incoming sequence numbers and maintain the most recently received sequence number. The receiver shall detect the end of a reassembled frame when it receives a fragment with the (E)nding bit set to 1. Reassembly of the frame shall be declared complete (and the frame delivered to the next network function) if all sequence numbers up to that fragment have been received.

The CLP indications from all the received frame header copies in all the received fragments of one frame shall be logically ORed. The resulting CLP indication shall be included in the reassembled frame header. EFCI shall be set in the reassembled frame header, equal to the value of EFCI in the last constituent fragment.

The receiver shall detect lost fragments on a given VC when one or more sequence numbers from that VC are skipped. When a lost fragment or fragments are detected on a VC, the receiver shall discard all currently incomplete frame reassembles, and subsequently received fragments, for that VC until it receives the next fragment that bears the (B)eginning bit. The fragment bearing the (B)eginning bit shall be used to begin reassembling a new frame.

In the event of an error (e.g., one or more fragments lost due to transmission error, or a reassembly buffer overflow), fragments which cannot be reconstructed back into the original frame shall be discarded by the receiver.

3.8.5 Fragment and Frame Sizes

This specification does not require nor recommend any specific fragment size. The fragment size is configured in the transmitter, and two peer transmitters need not use the same fragment size. The above fragmentation and reassembly procedures assure proper interoperability with different fragment sizes.

The maximum fragment size should be configured on a per-link basis. The optimal fragment size is the result of a tradeoff between the efficiency of large frames, the interface speed, and the required delay and delay variation characteristics of the applications. This is left to the discretion of the user.

Receivers must be able to reassemble complete frames up to the required maximum frame size specified in Section 1 of this specification.

4 Physical Layer

Frame-based ATM over SONET/SDH interfaces use SONET/SDH transport as full-duplex octet oriented synchronous links. Bit stuffing is not used; see section 4.4 for further details. The SONET/SDH interface characteristics documented in RFC 2615 [RFC2615] and G.707 (see Annex A) form the basis of this agreement to the extent detailed below.

4.1 Interface Format

An octet interface to the physical layer is used. There is no provision for sub-octets to be supplied or accepted. The octet stream is mapped into the SONET/SDH Synchronous Payload Envelope (SPE), with the octet boundaries aligned with the SPE octet boundaries. HDLC flags (01111110) shall be used for interframe fill to buffer out the asynchronous nature of the arrival of the HDLC framed signals.

Scrambling of HDLC framed signals is required to provide security against emulation of the SDH set-reset scrambler pattern and replication of the STM-N frame alignment word. The $x^{43} + 1$ self-synchronous scrambler is used following byte stuffing (see Section 4.4) during insertion into the SPE. The scrambler shall operate continuously through the bytes of the SPE, including the interframe fill, bypassing bytes of SONET/SDH Path Overhead and any fixed stuff (see Figure 20 of ANSI T1.105 [T1.105] or Figure 10-17 of G.707 [G.707]). The scrambling state at the beginning of a SPE shall be the state at the end of the previous SPE. Thus, the scrambler runs continuously and is not reset per frame. An initial seed is unspecified. Consequently, the first 43 transmitted bits following startup or reframe operation will not be descrambled correctly. The scrambler operates most significant bit first. The scrambler is fed as follows:

A[7], A[6], … A[0], B[7], B[6], etc…

where A and B are consecutive octets, and A[7] is the most significant bit of octet A. Note that this is the opposite bit order of the HDLC FCS calculation.

The transmitter operation is illustrated in Figure 8.

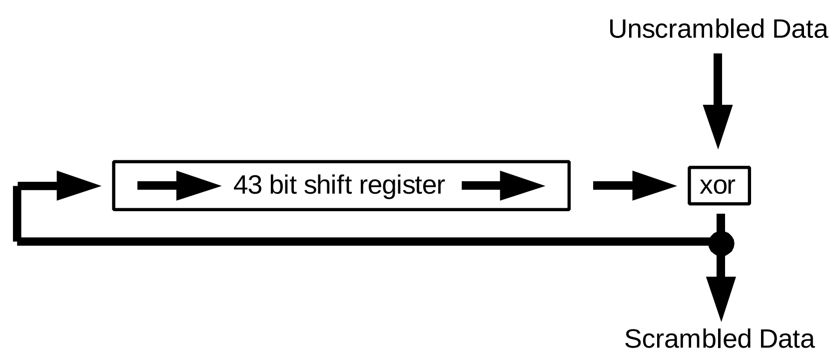

Figure 8 - Transmitter Schematic

The receiver operation is illustrated in Figure 9.

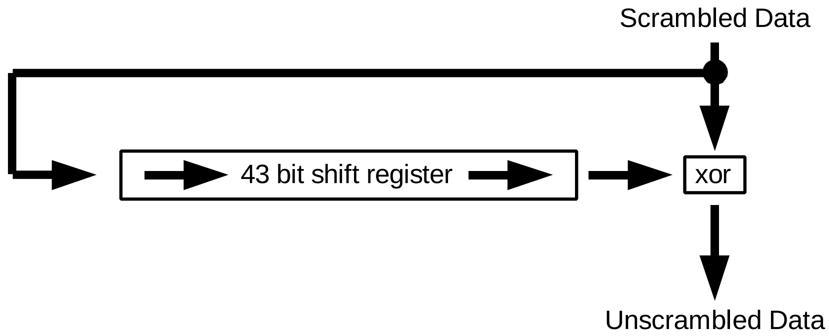

Figure 9 - Receiver Schematic

The Path Signal Label (C2) indicates the contents of the SPE. The value of 22 (0x16) is used to indicate a variable-length HDLC frame with scrambling enabled. Implementations must **not** use a Path Signal Label (C2) value of 207 (0xCF), which indicates a variable-length HDLC frame without scrambling.

The Multiframe Indicator (H4) is unused, and must be zero.

4.2 Transmission Rate

The basic rate for Frame Based ATM over SONET/SDH is that of STS-3c/STM-1 at 155.520 Mbps. The available information bandwidth is 149.760 Mbps, which is the STS-3c/STM-1 SPE with section, line and path overhead removed. This is the same super-rate mapping that is used for ATM and FDDI [T1.105].

Lower signal rates must use the Virtual Tributary (VT) mechanism of SONET/SDH. This maps existing signals up to T3/E3 rates asynchronously into the SPE, or uses available clocks for bit-synchronous and byte-synchronous mapping.

Higher signal rates must conform to the SDH STM series, rather than the SONET STS series. The STM series progresses in powers of 4 (instead of 3), and employs fewer steps, which simplifies multiplexing and integration.

4.3 Control Signals

The use of control signals is not required.

4.4 Framing and Octet Stuffing

The framing and octet stuffing for octet-oriented synchronous links are described in Section 4 of RFC 1662 [RFC1662], disregarding sections 4.4.2 and 4.5.2.

The ATM frames are located by row within the SPE payload. Because frames are variable in length, the frames are allowed to cross SPE boundaries.

4.5 HDLC Frame Check Sequence

Implementations must only support the 32-bit HDLC FCS to mitigate the error multiplication of this scrambler and to protect full-length Frame-based ATM over SONET/SDH frames. The CRC-32 generator polynomial is:

$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$ [RFC1662][I.363.5].

The 16-bit HDLC FCS is not used.

5 FAST Service Definition

The FAST Data Link Layer service is defined in this section, in terms of an abstract service model. That model is stated in terms of primitive functions and associated parameters. The model addresses the services provided by the FAST Data Link Layer and the AAL5 Common Part Convergence Sublayer (AAL5-CPCS).

5.1 Accessing the FAST Service

Upper layers may interact with the FAST Data Link Layer directly, or via the AAL5-CPCS. The former method must be used if the application wishes to receive and/or transmit OAM cells (see section 8), and by the mode 1 cell-to-frame interworking function (see section 3.1.4.2).

Figure 10 illustrates the relationships between the various layers. The underlined text identifies inter-layer primitive functions, and the text immediately below each primitive function lists the associated parameters.

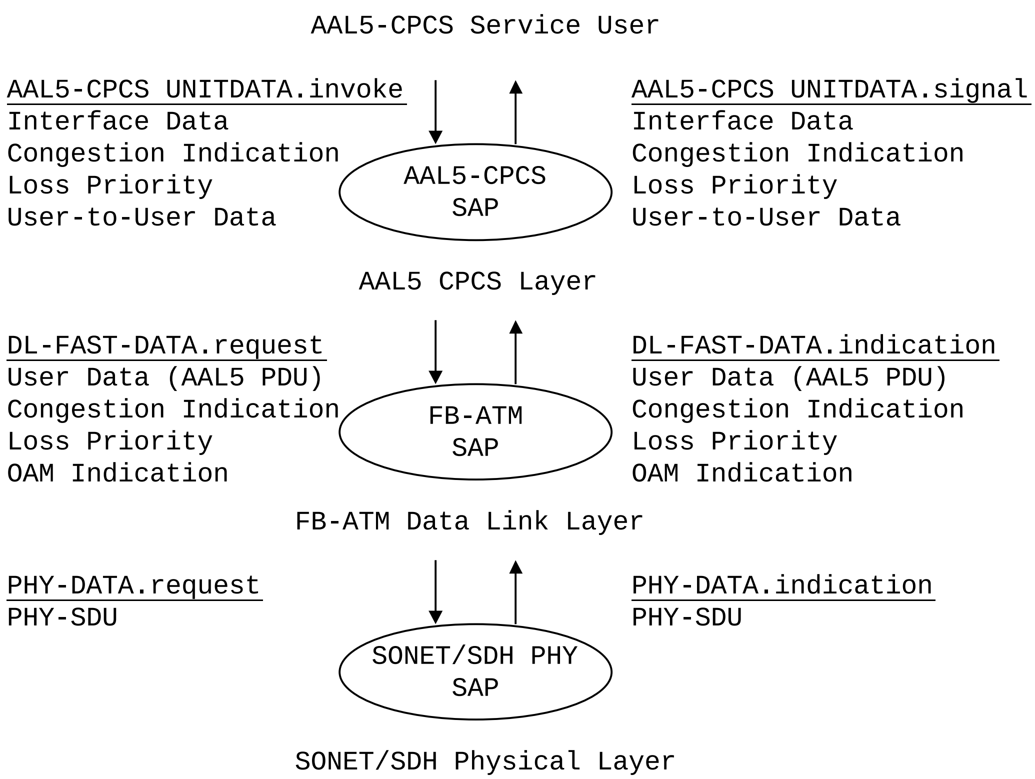

Figure 10 - Layer Inter-Relationships

5.2 FAST Data Link Layer Service Definition

The FAST Data Link layer utilizes AAL5 in message mode only, and does not utilize the corrupted data delivery option. An implementation may optionally support the transference of OAM frames to and from its ATM Layer Management (ATMM) layer (as indicated in section 3.6, the ability to transport OAM cells is mandatory at the FAST NNI and optional at the FAST UNI). If this capability is not supported, the FAST data link layer must discard OAM frames (frames in which the leftmost bit of the Payload Type field, is non-zero).

(R34) A FAST interface *shall* use AAL5 in message mode only and *shall not* use the corrupted data delivery option.

(O35) The FAST data link layer *may* support the transference of OAM frames to and from its ATMM layer.

(CR36) If the FAST data link layer does not support transference of OAM frames (as defined above) to and from its ATMM layer, it *shall* discard any OAM frames it receives.

The FAST data link layer service provides the primitives and parameters shown in Table 2. An "X" means a required parameter and an "O" means an optional parameter which is implemented only if the ability to transfer OAM frames to and from the ATMM layer, is implemented.

	DL-FAST-DATA.request	DL-FAST-DATA.indication
User Data	X	X
Congestion Indication	X	X
Loss Priority	X	X
OAM Indication	O	O

Table 2 - FAST Data Link Layer Primitives and Parameters

5.2.1 The DL-FAST-DATA.request Primitive

The DL-FAST-DATA.request primitive is used by the user of the FAST data link layer to access the FAST Service Access Point (FAST SAP) in order to pass a Service Data Unit (SDU) and other service parameters to the FAST data link Layer.

When invoking a DL-FAST-DATA.request, the following parameters are passed:

- **User Data**: the SDU passed via this parameter contains an AAL5-CPCS Protocol Data Unit (PDU) if the OAM Indication parameter is 0, and contains an OAM PDU (formatted as indicated in section 8) if the OAM Indication parameter is 1.
- **Congestion Indication**: This parameter is used by the FAST data link layer's user part to indicate local congestion. A value of 0 indicates no local congestion, a value of 1 indicates local congestion.
- **Loss Priority**: This parameter is used by the FAST data link layer's user part to indicate the relative priority of the SDU passed via the User Data parameter. A value of 0 indicates high priority, a value of 1 indicates low priority. Under some circumstances the network may preferentially discard low priority SDUs.
- **OAM Indication**: This parameter is used by the FAST data link layer's user part to indicate whether the SDU passed via the User Data parameter is to be interpreted as user data (i.e., an AAL5-CPCS PDU) or as an OAM PDU. A value of 0 indicates user data, a value of 1 indicates an OAM PDU.

(R37) The DL-FAST-DATA.request primitive *shall* be implemented as described in section 5.2.1.

5.2.2 The DL-FAST-DATA.indication Primitive

The DL-FAST-DATA.indication primitive is received by the user of the FAST data link layer at the FAST Service Access Point (FAST SAP) to allow the FAST data link layer to pass a Service Data Unit (SDU) and other service parameters to the FAST data link layer user.

When a DL-FAST-DATA.indication is received, the following parameters are passed:

- **User Data**: the SDU passed via this parameter contains an AAL5-CPCS Protocol Data Unit (PDU) if the OAM Indication parameter is 0, and contains an OAM PDU (formatted as indicated in section 8) if the OAM Indication parameter is 1.
- **Congestion Indication**: This parameter indicates the congestion status of the path between connection endpoints, in the incoming (toward the receiving FAST data link layer user) direction. A value of 0 indicates that no interface along this path, including the user part at the far end of the connection, has reported local congestion. A value of 1 indicates that some interface along this path has reported local congestion.
- **Loss Priority**: This parameter indicates the relative priority of the SDU passed via the User Data parameter. A value of 0 indicates high priority, a value of 1 indicates low priority. If locally congested, the FAST data link layer or its user part(s) may preferentially discard low priority SDUs.
- **OAM Indication**: This parameter indicates whether the SDU passed via the User Data parameter is to be interpreted as user data (i.e., an AAL5-CPCS PDU) or as an OAM PDU. A value of 0 indicates user data, a value of 1 indicates an OAM PDU.

(R38) The DL-FAST-DATA.indication primitive *shall* be implemented as described in section 5.2.2.

5.3 The AAL5 CPCS Service

Table 3 summarizes the primitives and parameters of the AAL5 CPCS service, as utilized when interfacing with the FAST data link layer. Normative requirements with respect to the implementation of the AAL5 Common Part Convergence Sublayer (CPCS) are found in ITU-T Recommendation I.363.5.

	AAL5-CPCS UNITDATA.invoke	AAL5-CPCS UNITDATA.signal
Interface Data	X	X
More (Note 1)	—	—
Loss Priority	X	X
Congestion Indication	X	X
User-to-User Information	X	X
Reception Status (Note 2)	—	—

Table 3 - AAL5 CPCS Service Primitives and Parameters

Note 1: Not present in message mode (which is the only mode supported by this specification)

Note 2: Associated with the corrupted data delivery option, which is not supported by this specification.

5.3.1 Mapping between FAST and AAL5 CPCS Primitives

This section defines the mapping of FAST primitives and parameters, to and from those of the AAL5 Common Part Convergence Sublayer (AAL5-CPCS).

5.3.1.1 Mapping the AAL5-CPCS UNITDATA.invoke to the FAST Data Link Layer

Upon receipt of an AAL5-CPCS UNITDATA.invoke signal, the AAL5 CPCS forms a DL-FAST-DATA.request primitive with the parameters noted below, and submits it to the FAST data link layer service.

- The FAST **User Data** parameter is formed as specified in section 5.2.1, using the Interface Data parameter of the AAL5-CPCS UNITDATA.invoke signal.
- The FAST **Congestion Indication** parameter is formed using the Congestion Indication parameter of the AAL5-CPCS UNITDATA.invoke signal.
- The FAST **Loss Priority** parameter is formed using the Loss Priority parameter of the AAL5-CPCS UNITDATA.invoke signal.
- The FAST **OAM Indication** parameter is set to 0, indicating that the data passed via the User Data parameter should be interpreted as user data, not as an OAM PDU.

Notes:

1. OAM data cannot be submitted to the FAST data link layer by way of the AAL5-CPCS layer. Upper layers that wish to transmit OAM information must access the FAST data link layer directly.
2. The AAL5-CPCS User-to-User Data parameter is placed into the trailer of the AAL5-CPCS PDU by the AAL5 CPCS. The result is passed to the FAST data link layer in the FAST **User Data** parameter.

(R39) OAM data *shall not* be submitted to the FAST data link layer by the AAL5-CPCS layer.

5.3.1.2 Mapping the DL-FAST-DATA.indication to the AAL5 CPCS

Upon receipt of a DL-FAST-DATA.indication primitive from the FAST data link layer, the AAL5 CPCS forms an AAL5-CPCS UNITDATA.signal with the parameters noted below, and submits it to the AAL5-CPCS service user.

– The AAL5-CPCS **Interface Data** parameter is obtained from the FAST User Data parameter by removing the trailer from the AAL5-CPCS PDU contained within the FAST User Data parameter.

– The AAL5-CPCS **Congestion Indication** parameter is obtained from the FAST Congestion Indication parameter.

– The AAL5-CPCS **Loss Priority** parameter is obtained from the FAST Loss Priority parameter.

– The AAL5-CPCS **User-to-User Data** parameter is obtained from the AAL5-CPCS trailer associated with the AAL5 PDU passed via the FAST User Data parameter.

Notes:

1. OAM data cannot be passed by the FAST data link layer to the AAL5-CPCS layer. Upper layers that wish to obtain OAM information from the network, must access the FAST layer directly.

(R40) OAM data *shall not* be submitted to the AAL5-CPCS layer by the FAST data link layer.

6 Signaling

This section identifies requirements related to the support of signaling at the FAST interface. Whether or not signaling is supported at a given FAST interface is (as it is with cell based ATM interfaces) determined by configuration and/or by the capabilities of the device.

6.1 Specification of the Signaling Channel

(R41) VPI/VCI values for the signaling channel *shall* be consistent with the UNI Signaling 4.0 specification (e.g., default to VPI=0, VCI=5).

6.2 Restrictions and Defaults

6.2.1 Connection Types

The FAST UNI may support switched virtual channel connections (SVCCs). As the FAST UNI does not support virtual path connections, it follows that it does not support switched virtual path connections (SVPCs). The FAST NNI may support either SVC or SVP connections, and in addition may support either / both SPVC and SPVP connections as defined in Annex C of PNNI 1.0 [PNNI 1.0].

(O42) Switched Virtual Channel Connections (SVCCs) *may* be supported at the FAST UNI.

(O43) Any combination of Switched Virtual Channel Connections (SVCCs), Switched Virtual Path Connections (SVPCs), Soft Permanent Virtual Channel Connections (SPVCCs) and Soft Permanent Virtual Path Connections (SPVPs) *may* be supported at the FAST NNI.

The signaling procedures originally defined in PNNI 1.0 in support of SPVC and SPVP connections, did not support AAL Parameters. They allowed, therefore, no standard way to determine whether an SPVCC utilizes AAL5 (and hence may be carried using frame encapsulation) or not. Implementations may carry such connections using cell encapsulation, or may support configuration options whereby the operator may indicate the adaptation layer which should be “inferred”. Such options, if implemented, are beyond the scope of this specification.

At this writing, the ATM Forum Control Signaling (CS) working group has agreed to relax the restrictions on Information Element usage defined PNNI 1.0 with respect to SPVC and SPVP connections. This will make it possible to include the AAL Parameters IE in the signaling messages used to establish SPVC connections, and hence to utilize frame encapsulation to transport AAL5 SPVCs without resorting to inference to determine its applicability.

(CR44) If the adaptation layer in use at the end points of a VCC traversing a FAST NNI is unknown, the FAST NNI *shall* either transport it using cell encapsulation, or implement a mechanism (not defined in this specification) which allows the desired encapsulation to be configured by the network operator.

6.2.2 ATM Adaptation Layer

A connection setup request initiated by a FAST UNI may only request AAL type 5 in the ATM Adaptation Layer Information Element.

(R45) Connection Setup requests initiated by a FAST UNI, *shall* specify AAL type 5.

(R46) Connection Setup requests received by a FAST UNI which request an unsupported AAL type, *shall* be rejected with Cause Code #93 (“AAL parameters cannot be supported”).

(R47) Connection Setup requests received by a FAST UNI which do not specify an AAL type, *shall* be processed as if they had specified AAL type 5.

A FAST interworking function must examine the adaptation layer information element to determine whether to utilize frame or cell encapsulation.

(R48) Connection Setup requests traversing a FAST IWF *shall* (if not rejected) result in the establishment of a bearer channel utilizing frame encapsulation, under the following circumstances:

- The request is for an SVCC, explicitly requests AAL type 5, and support of AAL5 VCCs of the requested type (e.g., service category) is consistent with the configuration of the IWF, or
- The request is for an SVCC or SPVCC, does not specify an AAL type, the IWF has been configured to infer “AAL5” in the absence of an explicitly indicated adaptation layer, and support of AAL5 VCCs of the requested type (e.g., service category) is consistent with the configuration of the IWF

(R49) Except in the circumstances outlined in the previous requirement, connection setup requests traversing a FAST IWF *shall* (if not rejected) result in the establishment of a bearer channel utilizing cell encapsulation.

6.2.3 Broadband Bearer Capability

Bearer Classes C and X[3] are supported at the FAST UNI. All bearer classes must be supported at the FAST IWF and at the FAST NNI.

(R50) Bearer Classes C and X *shall* be supported at the FAST UNI.

(R51) All bearer classes *shall* be supported at the FAST NNI, and at a FAST IWF.

(R52) Connection setup requests specifying an unsupported bearer class *shall* be rejected with Cause Code #65 (“bearer capability not implemented).

[3] Broadband Bearer Class X requires that an end-to-end cell relay connection between users be established. When carried over a FAST interface, such a connection may undergo cell-to-frame conversion and be subsequently transported as a frame. Despite this fact, class X is allowed in order to support existing applications which require use of BCOB-X. The IETF’s Classical IP over ATM specification (RFC 1577) is an example.

6.2.4 Traffic Management parameters

This specification places no constraints on the use of ATM Layer services and/or quality of service parameters, as defined in Traffic Management 4.1 and UNI Signaling 4.0. As with cell based interfaces, a FAST interface may implement all or a subset of the traffic management capabilities defined in these specifications.

6.3 Signaling Protocol Stack

Signaling protocols are supported without modification on FAST interfaces, using the protocol stack illustrated in Figure 10. The SSCF/SSCOP layer attaches to the AAL5-CPCS SAP as it would on a cell-based interface, using message mode and without use of the corrupted data delivery option.

7 ILMI

The Integrated Local Management Interface (ILMI) may be supported at FAST interfaces. If implemented, such support shall conform to the requirements stated in this section.

In general the fact that when operating over a FAST interface the underlying transport is frame-based rather than cell-based, is transparent to the ILMI application. The interactions between ILMI and the AAL5 Common Part Convergence Sublayer (CPCS) are identical to those used over cell-based interfaces. The interface between the AAL5 CPCS and the FAST Data Link Layer is specified in section 5.

(R53) The encapsulation of SNMP messages within AAL5, the interactions between ILMI and the AAL5 CPCS, and the ATM layer configuration (e.g., default VPI/VCI) *shall* be as specified in section 5.1 of [ILMI4.0].

(R54) The VCC used to transport ILMI messages over a FAST interface, *shall* utilize frame encapsulation.

8 OAM

8.1 Overview

This sub-section is informational.

Operations, Administration and Maintenance (OAM) cells are the means by which ATM Layer Management (ATMM) entities communicate across ATM networks. These entities provide abilities including but not limited to the following.

- fault management (alarm surveillance, loopback and continuity verification functions)
- performance management (activation/deactivation of performance monitoring for connections and connection segments, and associated reporting mechanisms)
- resource management (conveyance of network congestion status and per-connection rate adjustments)

ATM Layer Management entities reside within the ATM Management layer, which is a peer to the adaptation layer. There is, conceptually, an ATMM entity and an ATM entity per connection.

These relationships are illustrated in Figure 11. Note that at a FAST interface, the ATM entity is implemented within the FAST Data Link layer.

A path between a pair of ATMM entities is termed an *OAM flow*. The end points of an OAM flow may be coincident with those of the associated connection, or one or both may be at interfaces traversed by the connection but at which it does not terminate. The former is termed an *end to end* flow, and the latter a *segment* flow. At the end points of the OAM flow, OAM PDUs are delivered by the ATM entity to its ATMM peer (as illustrated in Figure 11). At other interfaces OAM data is transferred between interfaces by the ATM entity. This is analogous to the manner in which user data is transported between adaptation layer end points.

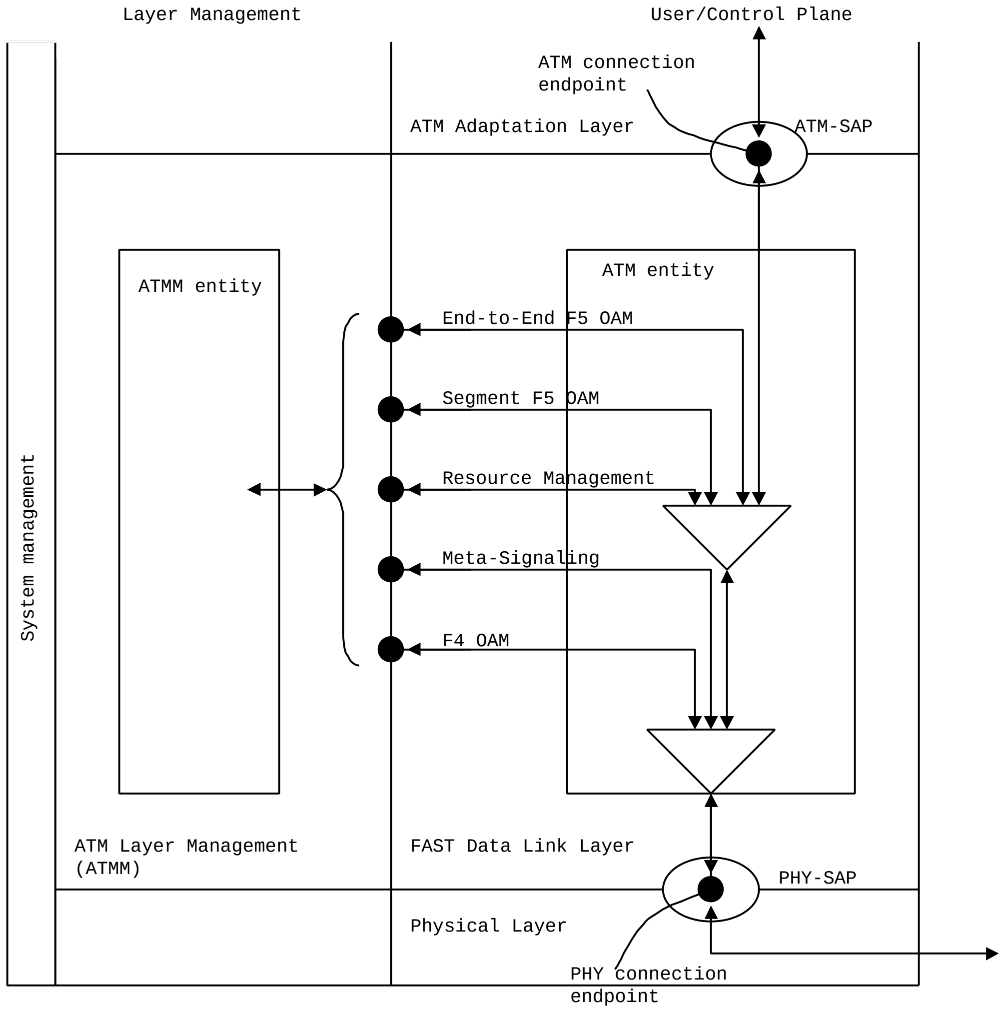

Figure 11 - Relationship of the ATMM and FAST Data Link Layers

The OAM Protocol Data Unit (PDU) is designed to fit into the payload of a single ATM cell. Its format is illustrated in Figure 12.

	8	7	6	5	4	3	2	1
1	OAM Type				Function Type			
2 . . . 46	Function Specific							
47	reserved						CRC-10	
48	CRC-10 (low-order 8 bits)							

Figure 12 - OAM Protocol Data Unit (PDU) Format

8.2 Encapsulation of OAM Cells

As indicated in section 3.6, OAM cells are transported over FAST interfaces via cell encapsulation. At an OAM flow end point this is accomplished by mapping the OAM PDU to the payload of a FAST frame, as specified in the following sub-sections. Requirements applicable to the FAST IWF are specified in section 8.4.

8.2.1 Mode 0 OAM Encapsulation Requirements

The result of encapsulating an OAM PDU within a FAST mode 0 frame, is illustrated in Figure 13.

	8	7	6	5	4	3	2	1
1	GFC (UNI) or VPI (NNI)				VPI			
2	VPI				VCI			
3	VCI							
4	VCI				PTI			CLP
5	OAM PDU (see Figure 12)							
.								
.								
.								
.								
.								
.								
52								
53	User to User Octet							

Figure 13 - OAM PDU encapsulated in FAST mode 0 frame

(R55) When encapsulating an OAM PDU within a FAST mode 0 frame, the User to User Octet is physically present but not utilized. Its value *shall* be set to zero by the transmitter and *shall not* be verified by the receiver.

8.2.2 Mode 1 OAM Encapsulation Requirements

The result of encapsulating an OAM PDU within a FAST mode 1 frame, is illustrated in Figure 14.

	8	7	6	5	4	3	2	1
1	GFC (UNI) or VPI (NNI)				VPI			
2	VPI				VCI			
3	VCI							
4	VCI				PTI			CLP
5	B	E	C	R	Sequence No. (high 4 bits)			
6	Sequence No. (low 8 bits)							
7 8	Cell Position Indicator (CPI)							
9 56	OAM PDU (see Figure 12)							

Figure 14 - OAM PDU encapsulated in FAST mode 1 frame

(R56) When encapsulating an OAM PDU within a FAST mode 1 frame, the trailer *shall* be omitted.

The Cell Position Indicator (CPI) field is set to zero in frames resulting from the receipt of OAM data from the local ATMM entity. This applies to the case where the OAM flow originates on the FAST mode 1 interface. An IWF may set it to values other than zero, as discussed in section 8.4.

(R57) The CPI field *shall* be set to zero in frames resulting from the receipt of OAM data from the local ATMM entity.

8.3 Inter-Layer Requirements

(R58) The ATMM Layer *shall* when passing OAM data to the FAST Data Link Layer, code the inter-layer parameters as follows:

- **User Data** shall contain an OAM PDU, formatted as illustrated in Figure 12
- **Congestion Indication** may be set to either 0 or 1.
- **Loss Priority** may be set to either 0 or 1
- **OAM Indication** shall be set to 1

(R59) The FAST Data Link Layer *shall* when passing OAM data to the ATMM Layer, code the inter-layer parameters as follows:

- **User Data** shall contain an OAM PDU, formatted as illustrated in Figure 12
- **Congestion Indication** may be set to either 0 or 1.
- **Loss Priority** may be set to either 0 or 1
- **OAM Indication** shall be set to 1

8.4 Interworking Requirements

A FAST IWF must support the transport of both user data and OAM data, between cell-based and frame-based interfaces. To do so it uses one of two mechanisms, defined previously as *cell encapsulation* and *frame encapsulation.* As indicated in section 3.6, OAM data must utilize the former.

To obtain OAM cells from a connection on a cell-based interface, the IWF interfaces directly to the associated ATM layer entity. Its mechanism for doing so is beyond the scope of this specification. It obtains frames from the FAST SAP as described in section 5.1.

In either mode 0 or mode 1, the first 4 bytes of the frame and the first 4 bytes of the corresponding cell, are identical (as specified in section 3.1.3.1). The HEC byte from the ATM cell is discarded by the IWF in the cell-to-frame direction, and generated in the frame-to-cell direction.

A mode 0 IWF creates the user to user octet in the cell-to-frame direction (setting its value to zero), and discard it in the frame-to-cell direction. This is consistent with the requirements specified in section 8.2.1.

When an OAM cell is received in the cell-to-frame direction at an IWF, it may be the case (if the associated connection utilizes frame encapsulation) that some number of cells belonging to the associated connection are enqueued, awaiting reassembly. Because the OAM cell cannot be reassembled, it must immediately be re-encapsulated (as described above) and queued for transmission.

(R60) When an OAM cell is received in the cell-to-frame direction by a FAST IWF, it *shall* immediately be re-encapsulated and scheduled for transmission (regardless of the reassembly state of the associated connection).

As a result of this requirement, the position of the OAM cell relative to the user data cells associated with the same connection, may change. This phenomenon is illustrated in Figure 15, where the two triangular objects represent cell-to-frame and frame-to-cell IWFs, respectively, and the flow of data is from left to right.

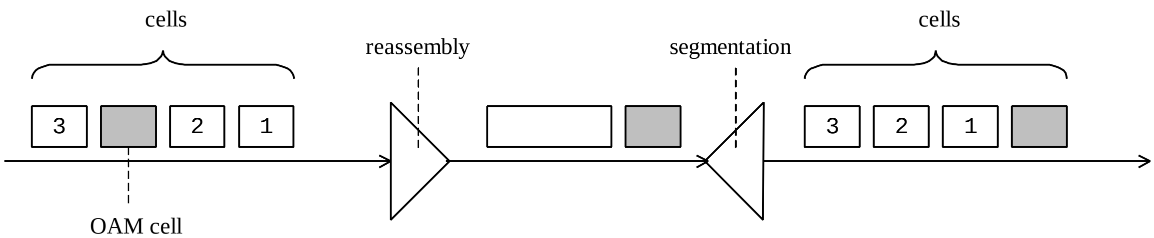

Figure 15 - OAM Cell Re-Positioning

Mode 0 provides no resolution to this issue. As such, it may not support OAM applications (e.g., performance monitoring) which are sensitive to the position of OAM cells within the sequence of user data cells belonging to the same connection.

A mode 1 IWF encodes the connection's the reassembly state into the Cell Position Indicator (CPI) field of the outgoing OAM frame in the cell to frame direction, and uses this information in the frame to cell direction to restore the position of the OAM cell relative to the associated user data cells.

(R61) When an OAM cell is received in the cell-to-frame direction by a mode 1 FAST IWF, it *shall* encode in the CPI field of the resulting frame, the number of cells of the associated connection currently queued for reassembly. This value *shall* be zero if the associated connection is being transported via cell encapsulation.

(R62) The sequence in which OAM cells for a given connection are transmitted (in the frame-to-cell direction) by the IWF *shall not* differ from that in which the corresponding OAM frames were received. For example if OAM frame A for connection X arrives at the IWF prior to the arrival of OAM frame B (also associated with connection X), the IWF *shall not* transmit the OAM cell corresponding to B sooner than it transmits the OAM cell corresponding to A.

This requirement prevents OAM cells associated with a given connection from being re-sequenced with respect to one another by the IWF, as might otherwise occur if for example they have the same CPI value.

(R63) When an OAM frame is received in the frame-to-cell direction by a mode 1 FAST IWF, the IWF *shall* transmit the corresponding OAM cell immediately after the first occurrence of any of the following:

- a number of user data cells from the associated connection equal to the value received in the CPI field of the OAM cell in question, has been transmitted.
- an OAM frame is received for the same connection, whose CPI value is less than the value received in the CPI field of the OAM frame from which the OAM cell in question, was generated.
- a user data cell from the same connection has been transmitted, whose SDU_type (rightmost bit of the PTI field) equals 1. Note that this is AAL5's means of communicating the end of the associated PDU.
- some implementation-specific limit (e.g., the maximum number of OAM cells which may be queued at the frame-to-cell IWF) has been exceeded.

The first condition will, in the absence of cell loss or implementation restrictions, move the OAM cell back to its original position within the associated sequence of user data cells. The second and third conditions are necessary in order to recover from upstream cell loss. The fourth reflects practical necessity. Implementers are advised, however, that the performance of OAM applications will suffer if such limits do not reflect the requirements of the network (e.g., number of connections, relative usage of OAM). While such requirements will vary, it is useful to set a lower bound which must be met by any implementation claiming compliance with this specification. This motivates a final requirement.

(R64) A mode 1 FAST IWF *shall* be capable of queuing at least 2 OAM cells per connection utilizing frame encapsulation, for a number of such connections judged suitable given the type and intended application of the interface, in the frame-to-cell direction, for the purpose of repositioning them within the associated sequence of user data cells.

9 Traffic Management

Note: The requirements in this section may be superseded by future Traffic Management specifications.

As indicated in section 3.4, FAST interfaces may simultaneously support connections belonging to multiple ATM layer service categories (e.g., CBR, rt-VBR, nrt-VBR, etc.,). The ability to do so, while maintaining the desired service quality, requires the following:

(a) A definition of conformance applicable to the frames of a given connection, which is a part of the associated traffic contract, and

(b) A set of capabilities within the network, sufficient to guarantee the desired quality of service to that traffic which satisfies the conformance requirements of the associated service.

As with cell-based interfaces, the details of (b) are not subject to standardization.

(R65) The quality of service provided to a connection crossing a FAST interface, *shall* be consistent with that indicated, implicitly or explicitly, by the QoS parameters associated with that connection.

This specification addresses point (a) by defining the manner in which conformance may be verified at a FAST interface.

9.1 Conformance Definition

9.1.1 Source Traffic Descriptor

The source traffic descriptor associated with a connection originating at or spanning a FAST interface, is identical both syntactically and semantically to that defined for use at cell-based interfaces [TM4.1]. This includes specification of rates and tolerances in terms of cells.

9.1.2 Frame Connection Compliance

The compliance definition requirements specified in section 4.3 of [TM4.1] are applicable to connections which originate at or span, FAST interfaces.

9.1.3 Conformance Definition

Appendix VI.2 of [TM4.1] defines a "simple frame-based GCRA" (Simple F-GCRA) which may with the modification described below, be applied at FAST interfaces, in any case where (for a cell based interface) use of the GCRA would be applicable. The values which would per [TM4.1] be used for the (I)ncrement and (L)imit parameters of the GCRA, are used for the T and L parameters, respectively, of the Simple F-GCRA.

In order to apply the Simple F-GCRA at a FAST interface, it is necessary to compute the number of cells into which the FAST PDU would have been sub-divided, had it arrived on an ATM cell interface. For connections utilizing cell encapsulation this result is always 1. For connections utilizing frame encapsulation, it is derived as follows:

$$number\_of\_cells(packet_i) = \lceil (payload\_size(packet_i) + 8) \div 48 \rceil \qquad (1)$$

Where:

$\lceil x \rceil$	refers to the smallest integer greater than or equal to x, and
payload_size	is the length of the User Data portion of the FAST information field (as illustrated in Figure 5 and Figure 6).

When applied at a FAST interface, the test defined in Part 1 of TM4.1 Appendix VI.2 is slightly modified. The resulting algorithm is illustrated in its entirety, in Annex B.

Conformance may then be defined at the FAST interface using the Simple F-GCRA defined in Annex B, by treating each arriving $packet_i$ as a sequence of $number\_of\_cells(packet_i)$ cells, all with the same CLP marking, arriving simultaneously (i.e., with the same arrival time t_a).

9.1.3.1 Definition of Conformance for CBR, rt-VBR, nrt-VBR and UBR services

(R66) Conformance of connections originating at or spanning FAST interfaces and associated with the CBR, rt-VBR, nrt-VBR or UBR services, *shall* be as specified in section 4.5 of [TM4.1], substituting the Simple F-GCRA defined in Annex B for the GCRA as indicated in section 9.1.3.

9.1.3.2 Definition of Conformance for the GFR service

(R67) Conformance of connections originating at or spanning FAST interfaces and associated with the GFR service, *shall* be as specified in section 4.5 of [TM4.1], substituting the Simple F-GCRA defined in Annex B for GCRA as indicated in section 9.1.3 and applying the MFS test to the result of the number_of_cells function defined in equation (1) above.

Service eligibility for GFR connections spanning FAST interfaces is defined as specified in section 4.5.5.2 of [TM4.1]. The technique outlined in section 9.1.3 is used to “emulate” the arrival of a sequence of cells equivalent to the FAST PDU.

9.1.3.3 Definition of Conformance for the ABR service

As indicated in section 4.5.4 of [TM4.1], the definition of conformance for the ABR service is network specific. The possible need for and specification of an ABR conformance definition specific to FAST interfaces, is for further study.

9.2 Translation of Units

For compatibility with existing ATM protocols, and to facilitate interworking with cell-based interfaces, traffic parameters associated with connections originating at or spanning FAST interfaces are expressed in units of cells. This section discusses the applicability of these parameters at FAST interfaces.

9.2.1 Traffic Parameters

As indicated in section 9.1, conformance at FAST interfaces is also defined in terms of cells. Therefore it is unnecessary to convert traffic parameters from units of cells or cells/second, in order to verify conformance at a FAST interface.

Whether it is necessary to adjust traffic parameters to perform Connection Admission Control or resource allocation, is beyond the scope of this specification. Note that neither procedure is subject to standardization.

9.2.2 Link Attributes

PNNI advertises attributes about network resources (e.g., links) which are used to select paths. Examples include resource availability (e.g., Maximum Cell Rate and Available Cell Rate) and estimated impairments (e.g., Cell Transfer Delay and Cell Delay Variation). The derivation of the values of these parameters, is not subject to standardization.

Recommendations with respect to potential adjustment to the parameters advertised by PNNI with respect to a FAST interface, are for further study.

9.2.3 Performance Verification

Reference architectures, performance measures and verification mechanisms are defined in various ITU-T recommendations and ATM Forum specifications. All explicitly presume cell-based interfaces. Production of equivalent text applicable to FAST interfaces, is for further study.

9.2.4 Accommodation of “Octet Stuffing” at the Physical Layer

As noted in section 4.4, the FAST physical layer implements octet stuffing in order to achieve data transparency. In the transmit direction, after computing the FCS and prior to scrambling, the physical layer replaces any “special characters” which occur within the FAST PDU with a 2-octet “escape sequence”. Such a sequence starts with a Control Escape octet (`0x7d`), followed by an octet whose value is derived from the value of the “special character”. In the receive direction, after unscrambling, the physical layer replaces any two-byte sequence beginning with the Control Escape octet with a single octet whose value is derived from the second octet in the sequence. The net effect is, of course, to restore the octet which was “escaped” by the transmitter.

This process is repeated link by link, and at each link it may differ (since each link may have its own definition of what constitutes a special character). At minimum each link must (per [RFC1662]) recognize as special characters the HDLC flag byte (`0x7E`) and the Control Escape character (`0x7D`).

The amount of bandwidth lost to octet stuffing depends on the frequency with which special characters occur within the data stream. If a link supports only the mandatory control sequences, and assuming that these occur with no greater or lesser frequency than any other bit pattern, the increase in bandwidth required will on average be `0.78%` (`2/256`). In the worst case (packets comprised entirely of special characters) the increase will be nearly `100%`.

Because octet stuffing is added and removed at the FAST physical layer, it has no effect on the packet arrival rate measured at higher layers. Hence, rates of individual connections need not be adjusted to account for octet stuffing.

Effectively, octet stuffing reduces the bandwidth of the link. Networks may wish to adjust related capacities (for example, the Maximum Cell Rate resource attribute advertised by PNNI) by an equivalent amount.

10 References

[RFC1662] IETF RFC 1662, “PPP in HDLC-like Framing”, W. Simpson (ed.), July 1994.

[T1.105] ANSI T1.105, “Synchronous Optical Network (SONET) Basic Description including Multiplex Structure, Rates, and Formats”, American National Standards Institute, New York, 1995.

[G.707] ITU Recommendation G.707, “Network node interface for the synchronous digital hierarchy (SDH)”, ITU, Geneva, 1996.

[RFC2615] IETF RFC 2615, “PPP over SONET/SDH”, A. Malis and W. Simpson (ed.), June 1999

[I.363.5] ITU Recommendation I.363.5, “B-ISDN ATM Adaptation Layer specification: Type 5 AAL”, ITU, Geneva, 1996

[G707.2000] ITU Recommendation G.707, “Network node interface for the synchronous digital hierarchy (SDH)”, ITU, Geneva, 2000 (*final approval anticipated in October 2000*).

[DXI] ATM Forum af-dxi-0014.000, ATM Data eXchange Interface (DXI) Specification, Version 1.0, August , 1993

[FUNIv1] ATM Forum af-saa-0031.000, Frame based User-to-Network Interface (FUNI) Specifications V1.0, September, 1995

[FUNIv2] ATM Forum af-saa-0088.000, Frame based User-to-Network Interface (FUNI) Specifications V2.0, July, 1997

[FUNI.mse] ATM Forum af-saa-0109.000, Multi-service Extensions to FUNI v2.0 Specification, February 1999

[ILMI4.0] ATM Forum af-ilmi-0065.000, Integrated Local Management Interface (ILMI) Specification Version 4.0, September 1996

[RFC1990] IETF RFC 1990, "The PPP Multilink Protocol (MP)", K. Sklower, B. Lloyd, G. McGregor, D. Carr, T. Coradetti (ed), August 1996.

[TM4.1] ATM Forum af-tm-0121.000, Traffic Management Specification Version 4.1, September 1999

[PNNI1.0] ATM Forum af-pnni-0055.000, Private Network-Network Interface Specification, Version 1.0, March 1996

Annex A G.707 (2000)

This Annex contains text excerpted from G.707/2000 [G707.2000], that was determined at the April 2000 meeting of ITU-T Study Group 15. G.707/2000 is expected to be submitted to the ITU-T World Telecommunications Standardization Assembly (thereby completing the ITU-T approval process) in October 2000.

9.3.1.3 Signal label: C2

One byte is allocated to indicate the composition or the maintenance status of the VC-4-Xc/VC-4/VC-3. Table 9-11 below, which is based on Hex code, provides codes for this byte.

Table 9-11/G.707 - C2 byte coding

MSB 1 2 3 4	LSB 5 6 7 8	Hex code (Note 1)	Interpretation
0 0 0 0	0 0 0 0	00	Unequipped or supervisory-unequipped (Note 2)
0 0 0 0	0 0 0 1	01	Reserved (Note 3)
0 0 0 0	0 0 1 0	02	TUG structure, see 7.2
0 0 0 0	0 0 1 1	03	Locked TU-n (Note 4)
0 0 0 0	0 1 0 0	04	Asynchronous mapping of 34 368 kbit/s or 44 736 kbit/s into the Container-3, see 10.1.2
0 0 0 0	0 1 0 1	05	Mapping under development (Note 9)
0 0 0 1	0 0 1 0	12	Asynchronous mapping of 139 264 kbit/s into the Container-4, see 10.1.1.1
0 0 0 1	0 0 1 1	13	ATM mapping, see 10.2.1 and 10.2.2
0 0 0 1	0 1 0 0	14	MAN DQDB [1] mapping, see 10.4
0 0 0 1	0 1 0 1	15	FDDI [3]-[11] mapping, see 10.5
0 0 0 1	0 1 1 0	16	Mapping of HDLC/PPP [12], [13] framed signal according to 10.3
0 0 0 1	0 1 1 1	17	Mapping of Simple Data Link (SDL) with SDH self synchronising scrambler (Note 8)
0 0 0 1	1 0 0 0	18	Mapping of HDLC/LAPS [15] framed signals according to 10.3
0 0 0 1	1 0 0 1	19	Mapping of Simple Data Link (SDL) with set-reset scrambler (Note 8)
0 0 0 1	1 0 1 0	1A	Mapping of 10 Gbit/s Ethernet frames [14] (Note 8)
0 0 0 1	1 0 1 1	1B	Flexible Topology Data Link mapping (Note 8)
1 1 0 0	1 1 1 1	CF	Reserved (Note 7)
1 1 1 0 … 1 1 1 1	0 0 0 1 … 1 1 0 0	E1 … FC	Reserved for national use.
1 1 1 1	1 1 1 0	FE	Test signal, O.181 specific mapping (Note 5)
1 1 1 1	1 1 1 1	FF	VC-AIS (Note 6)

NOTE 1 – There are 209 spare codes left for future use.

NOTE 2 – Value "0" indicates "VC-4-Xc/VC-4/VC-3 path unequipped or supervisory-unequipped". This value is originated in the case of an open connection and in the case of a supervisory unequipped signal that contains no payload.

NOTE 3 – Value "1" should not be used in any equipment designed after the approval (10/00) of this recommendation. In the past this code meant "Equipped – non-specific" and has been used in cases where a mapping code is not defined in the above table, see code "05" for new designs. For interworking with (old) equipment designed to transmit only the values "0" and "1", the following conditions apply:

– for backward compatibility, old equipment shall interpret any value received other than value "0" as an equipped condition;

– for forward compatibility, when receiving value "1" from old equipment, new equipment shall not generate a PayLoad Mismatch alarm.

NOTE 4 – The code "03" shall, for backward compatibility purposes, continue to be interpreted as previously defined even if the locked mode byte synchronous mappings are not defined any more.

NOTE 5 – Any mapping defined in Recommendation O.181 which does not correspond to a mapping defined in Recommendation G.707 falls in this category.

NOTE 6 – Value "FF" indicates VC-AIS. It is generated by a TCM source if no valid incoming signal is available and a replacement signal is generated.

NOTE 7 – Previous value assigned for an obsolete mapping of HDLC/PPP framed signal [12], [13].

NOTE 8 – These mappings are under study and the signal labels provisionally allocated.

NOTE 9 – Value "05" is only to be used in cases where a mapping code is not defined in the above table. By using this code the development or experimental activities is isolated from the rest of the SDH network. There is no forward compatibility if a specific signal label is assigned later. If that is done the equipment that has used this code must either be reconfigured to use that new specific signal label or be recycled.

10.3 Mapping of HDLC framed signals

The mapping of HDLC framed signals [2] is performed by aligning the byte structure of every frame with the byte structure of the Virtual Container used including the concatenated structure (VC-n-Xc/VC-n-Xv/VC-n). Since the HDLC frames are of variable length (the mapping does not impose any restrictions on the maximum length) a frame may cross the Container-x frame boundary.

HDLC flags (01111110) shall be used for interframe fill to buffer out the asynchronous nature of the arrival of the HDLC framed signals according to the effective payload of the Virtual Container used (this excludes any fixed stuff bytes).

The HDLC framed signal plus the interframe fill shall be scrambled before they are inserted as payload of the Virtual Container (VC-n-Xc/VC-n-Xv/VC-4/VC-3) used. In the reverse operation, following termination of the VC signal, the payload will be descrambled before it is passed on to the HDLC layer. A self synchronizing scrambler with generator polynomial $x^{43}+1$ shall be used.

The $x^{43}+1$ scrambler shall operate continuously through the bytes of the VC-n-Xc/VC-n-Xv/VC-4/VC-3, bypassing bytes of SDH Path Overhead. The scrambling state at the beginning of a VC-n-Xc/VC-n-Xv/VC-4/VC-3 shall be the state at the end of the previous VC-n. Thus, the scrambler runs continuously and is not reset per frame. An initial seed of the scrambler is unspecified. Consequently, the first 43 transmitted bits following startup or SDH reframe operation will not be descrambled correctly.

The $x^{43}+1$ scrambler operates on the input data stream with Most Significant Bit (MSB) first, consistent with the bit ordering and transmission ordering defined for SDH in clause 5.

The above mapping procedure with scrambling shall be used for the mapping of HDLC framed signals (e.g. HDLC/PPP or HDLC/LAPS with IP packets) in any VC-n-Xc/VC-n-Xv/VC-4/VC-3 while scrambling is not required for VC-2/VC-12/VC-11.

There are no further specific requirements for any Virtual Container size, other than that the appropriate signal label for that Container is inserted in the appropriate Path Overhead location. Path signal labels are specified in paragraph 9.3.

Annex B. Frame-based GCRA (F-GCRA)

This section defines the UPC mechanism referenced in section 9.1 of this specification. It is a slightly modified version of the Simple F-GCRA(T, L) defined in Appendix VI.2 of [TM4.1]. The modification, which is required in order to remove dependence on CLP setting, is illustrated with strike-throughs.

Part 1: Arrival of the first cell of a frame at time t_a at a given UNI or inter-network interface on the ATM connection:

```
X' := X - (t_a - LPT)

if (X' > L) ~~or (the cell has CLP=1)~~ then
      passed := false
else
      passed := true
endif

if (passed) then
      X := max(0, X') + T
      LPT := t_a
endif
```

Part 2: Arrival of subsequent cells of a frame at time t_a at a given UNI or inter-network interface on the ATM connection:

```
if (passed) then
      X' := X - (t_a - LPT)
      X := max(0, X') + T
      LPT := t_a
endif
```

In the above algorithm `X` denotes the value of the Leaky Bucket Counter and `LPT` is the Last Pass Time. `X'` and `passed` are auxiliary variables. At the time of arrival t_a of the first cell of the connection to cross the given interface, `X = 0` and `LPT` $= t_a$.

Optimization

Because all cells are assumed to have the same arrival time t_a , Part 2 can be eliminated and Part 1 simplified. The resulting algorithm is as follows:

```
X' := X - (t_a - LPT)

if (X' > L) ~~or (the cell has CLP=1)~~ then
      passed := false
else
      passed := true
      X := max(0, X') + T * number_of_cells(packe_ti)
      LPT := t_a
Endif
```

Where *number_of_cells()* is defined as in Equation (1) of section 9.1.3.

End of Document