The ATM Forum
Technical Committee

Frame Based User-To-Network Interface (FUNI) Specification v2.0

AF-SAA-0088.000

July, 1997

The information in this publication is believed to be accurate as of its publication date. Such information is subject to change without notice and The ATM Forum is not responsible for any errors. The ATM Forum does not assume any responsibility to update or correct any information in this publication. Notwithstanding anything to the contrary, neither The ATM Forum nor the publisher make any representation or warranty, expressed or implied, concerning the completeness, accuracy, or applicability of any information contained in this publication. No liability of any kind shall be assumed by The ATM Forum or the publisher as a result of reliance upon any information contained in this publication.

The receipt or any use of this document or its contents does not in any way create by implication or otherwise:

- Any express or implied license or right to or under any ATM Forum member company's patent, copyright, trademark or trade secret rights which are or may be associated with the ideas, techniques, concepts or expressions contained herein; nor

- Any warranty or representation that any ATM Forum member companies will announce any product(s) and/or service(s) related thereto, or if such announcements are made, that such announced product(s) and/or service(s) embody any or all of the ideas, technologies, or concepts contained herein; nor

- Any form of relationship between any ATM Forum member companies and the recipient or user of this document.

Implementation or use of specific ATM standards or recommendations and ATM Forum specifications will be voluntary, and no company shall agree or be obliged to implement them by virtue of participation in The ATM Forum.

The ATM Forum is a non-profit international organization accelerating industry cooperation on ATM technology. The ATM Forum does not, expressly or otherwise, endorse or promote any specific products or services.

NOTE: The user's attention is called to the possibility that implementation of the ATM interoperability specification contained herein may require use of an invention covered by patent rights held by ATM Forum Member companies or others. By publication of this ATM interoperability specification, no position is taken by The ATM Forum with respect to validity of any patent claims or of any patent rights related thereto or the ability to obtain the license to use such rights. ATM Forum Member companies agree to grant licenses under the relevant patents they own on reasonable and nondiscriminatory terms and conditions to applicants desiring to obtain such a license. For additional information contact:

The ATM Forum
Worldwide Headquarters
2570 West El Camino Real, Suite 304
Mountain View, CA 94040-1313
Tel: +1-415-949-6700
Fax: +1-415-949-6705

Table of Contents

List of Figures

List of Tables

This page is intentionally blank

1.0 SCOPE

This document fully defines and specifies a Frame-based User-to-Network Interface which is known by the acronym FUNI.

The FUNI is capable of transparently supporting Service Specific Convergence Sublayers (SSCS) and other higher layers. However, the use and support of SSCSs - other than those supporting the UNI functionality, such as signaling - is beyond the scope of this document.

The only ATM adaptation layers supported on this interface are AAL5 and AAL3/4.

The FUNI is capable of supporting the basic ATM UNI functions (per ATM UNI 3.1) such as VPI/VCI multiplexing, signaling, network management, traffic shaping, and, optionally, OAM functions. The support of some of these functions may be limited in comparison to the full functionality of an ATM UNI, e.g., the FUNI does not support some types of traffic, therefore, it does not support the corresponding signaling and traffic parameters. Specific details are addressed in the specific requirements as follows.

This interface is intended to support only UNI 3.1 VBR and UBR type services, and TM 4.0 nrt-VBR and UBR traffic classes.

This document is intended to fully cover the FUNI specifications, thus entirely replacing the first version of the original FUNI specifications [af-saa-0031.000]. It should be noted, however, that an interface that supports [af-saa-0031.000] will remain compatible with this document's specifications at the user plane. This does not present any compatibility issues since [af-saa-0031.000] did not specify the management and control plane implementation agreements.

This document's specifications offer the following enhancements over [af-saa-0031.000].

1. Clear definition of signaling over a FUNI.
2. Support of the communication protocol of ILMI across a FUNI.
3. Application of Usage Parameter Control to connections across a FUNI
4. Handling of OAM cells across a FUNI including those that arrive during the reassembly of a user frame.
5. A Management Information Base (MIB) that enables configuration and management of a FUNI.
6. Optional support of multiple FUNI interfaces over a single DS1/E1 physical layer is permitted.
7. Optional support of alternative Physical Layers is permitted.
8. Removal of reference to VP connections, which are *not* supported by the FUNI interface.

It should be noted that item 6 above does not constitute any change to the original FUNI protocol - which is capable of supporting item 6 - but, it removes the explicit implementation restriction in the original specification. The purpose of adding item 6 in this document is to explicitly indicate that the support of multiple FUNI interfaces over a single DS1/E1 is an option available to the implementors without a need for changing the original protocol. Furthermore, the MIB specified in this document supports that option. The same is applicable to item 7, i.e., the FUNI documents explicitly specify DS1/E1 physical interfaces without excluding others. In this document the MIB contains a provision to support "other" physical interfaces which become the choice of the implementor who is also responsible for providing the proper management information base in the form of an enterprise MIB to support such optional physical interface(s), i.e., the full MIB supporting any optional physical interface chosen by an implementor is beyond the scope of this document.

It should also be noted that the user plane as specified and described in the original document [af-saa-0031.000] is not changed. Except for the AAL5 CPCS Services described in Section 5.4 below, all additions in this document are concerned with the control or management planes. AAL5 CPCS services in Section 5.4 cover user plane as well as control plane. This is not a change in the user plane, it is rather a clarification.

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

2.0 REFERENCE MODEL

Figure 2.1 depicts the functional reference model of the FUNI specified in this document. The user's FUNI payload carried in the DS1/E1 physical layer may be the full DS1/E1 line bandwidth or a fraction thereof. The specifications in this document treat any fractional DS1/E1 bandwidth, i.e., Nx64 Kbps where N is less than 24 for DS1 and less than 31 for E1, as a single payload/single FUNI interface. This means that all of the Nx64 time slots are treated as concatenated bandwidth and not separate DS0 channels. In the optional implementations which contain more than one FUNI interface (more than one Nx64 logical connection) in the same DS1/E1 physical interface, the multiplexing and demultiplexing of such FUNI interfaces over the physical interface is beyond the scope of this document.

It should be noted that the FUNI interface enables its users to communicate with users employing the same type of interface or a native ATM interface as shown in Figure 2.1 on the right hand side of the diagram.

The connection between a user and the network may be a logical connection which is part of a higher bandwidth physical connection. Figure 2.2 shows two examples: example (a) represents a full or fractional DS1 FUNI payload carried in a DS3 physical connection; and example (b) represents a fractional DS1 FUNI payload carried in a DS1 physical connection. The latter example represents a particularly useful configuration for users who wish to carry payloads other than the FUNI, e.g., voice, or multiple FUNI interfaces in the same DS1 physical connection to the (carrier) network. As seen from the diagrams, each FUNI payload is groomed as a separate interface into the ATM conversion function supporting it. Other payloads are groomed to the appropriate network element(s). Although the multiplexing and demultiplexing/grooming methods are beyond the scope of this document, it should be noted that employing traditional TDM technology and standards is the predominant method of achieving this function.

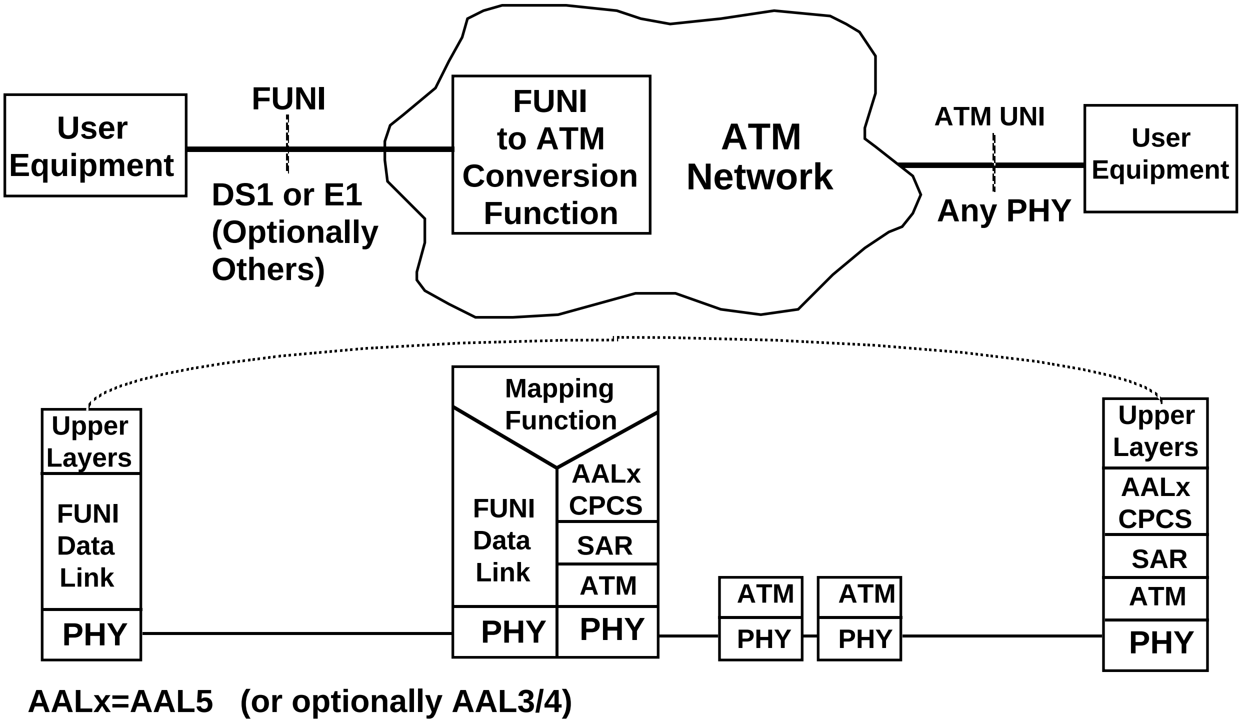


Figure 2.1 Reference Model for the FUNI

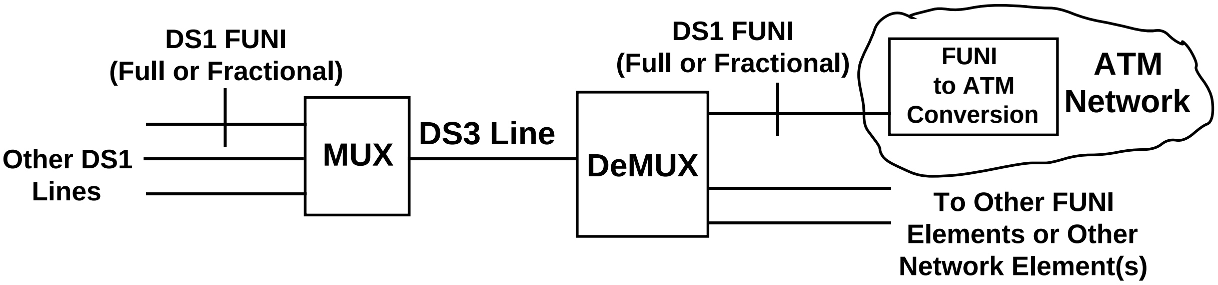

DS1 FUNI (Full or Fractional)
Other DS1 Lines
MUX
DS3 Line
DeMUX
DS1 FUNI (Full or Fractional)
FUNI to ATM Conversion
ATM Network
To Other FUNI Elements or Other Network Element(s)

Figure 2.2a Example of Multiplexing/Grooming a DS1 FUNI

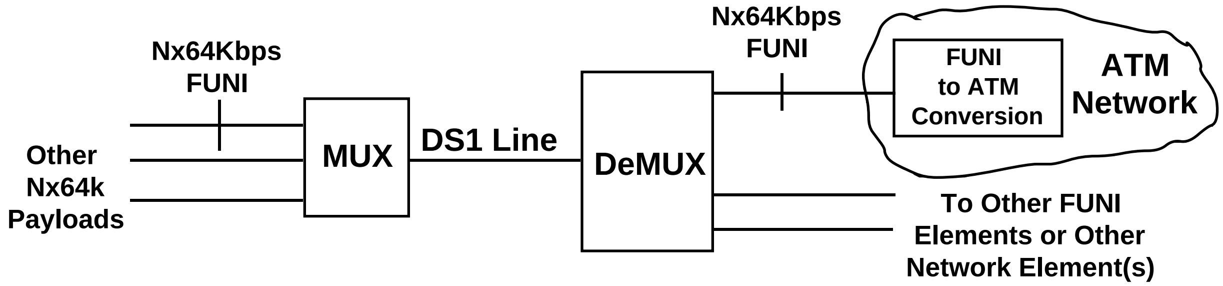


Figure 2.2b Example of Multiplexing/Grooming a Fractional DS1 FUNI

(R) Equipment claiming conformance to this specification *Shall* support at least one FUNI interface of Nx64 Kbps bandwidth with N up to 24 for DS1 and up to 31 for E1 physical interfaces respectively. Equipment supporting DS1 or E1 or both is conformant to this requirement.

(O) Conformant equipment may support any physical layer interface(s) in separate or concatenated bandwidth increments representing one or multiple FUNI interface(s) of Nx64k each. In this case N may exceed 31. Multiplexing and demultiplexing the bandwidth increments is beyond the scope of this document.

3.0 DATA LINK LAYER PROTOCOL

In order to avoid duplication of information and/or possible confusion, this section is organized as follows. The requirements and options are stated by functions and are given a numerical order (R1, R2, etc.; O1, O2, etc. and CR1, CR2, etc.). The modes of operation are shown in Table 3.1 below and each applicable R, O or CR are indicated next to the mode of operation. The modes of operation supported by each of the requirements, options and conditional requirements are stated next to each of these items in remainder of this section. Note that there is no mode of operation "Mode 2," this is to avoid confusion or conflict with Mode 2 of the ATM DXI.

Mode of Operation	**Summary**	**Applicable Requirements**
1a, Required	2 Octet Header, 4096 Octet SDU, 2 Octet CRC, AAL5 only, ≤ 512 user VCCs, 16 Reassembly, ≤ **DS1/E1**	R0, R1, R2, R3, R4, O3, CR1, R6, R8, R9, R10, R11, R12, R13
1b, Optional	1a plus AAL3/4 Options/CRs	1a plus O2, CR2, CR3
3, Required	2 Octet Header, 9232 Octet SDU, 4 Octet CRC, AAL5 only, ≤ 512 user VCCs, ≥ **DS1/E1**	R0, R1, R2, R3, R5, O3, CR1, R7, R8, R9, R10, R12, R13
4, Optional	4 Octet Header, 9232 Octet SDU, 4 Octet CRC, AAL5 only, Up to full count of usable VCCs, ≥ **DS1/E1**	R0, O1, R2, R3, R5, O3, CR1, R7, R8, R9, R10, CR4, CR5

Table 3.1 Modes of Operation and Their Requirements

(R0) The FUNI data link protocol encapsulation *Shall* conform to the following ITU-T recommendations

- Section 2.2 of Q.921, Flag sequence
- Section 2.6 of Q.921, Transparency
- Section 2.7 of Q.921, Frame Check Sequence for 16 bit CRC
- Section 6.3.2.1.2 of I.363 Addendum 1, AAL5 CPCS Structure and Coding, 32 bit CRC

All modes of operation

3.1 Frame Format and Frame Header Size

(R1) The FUNI *Shall* support a 2 octet frame header (based on ITU-T recommendation Q.922 Section 3.2). The frame and its header formats *Shall* meet Section 3.8 requirements in this document.

Modes 1a, 1b and 3.

(O1) The FUNI May support a 4 octet frame header (based on ITU-T recommendation Q.922 Section 3.2). The frame and its header formats *Shall* meet Section 3.9 conditional requirements in this document.

Mode 4 Only.

3.2 Number of Virtual Connections per FUNI

Each FUNI with 2 octet frame header is capable of supporting up to 512 user VCCs using combinations of the 16 VPI values inherent in the FA and the 32 VCI values available with each of the VPIs.

Each FUNI with 4 octet frame header is capable of supporting up to 256x(2^{16} - 32) user VCCs.

*Note that for all VPI values, VCI values 0 to 31 (32 VCCs) are reserved and must not be used for user data***.**

(R2) For each FUNI interface, equipment implementors *Shall* declare the maximum number of VCCs available to user and the maximum VPI and VCI values supported; and the user *Shall* notify the network operator of those values at time of subscription. The equipment *Shall* follow ITU-T recommendation I.361 (11/95) Section 2.2.3 rules in allocating the bits in the VPI and VCI fields.

All Modes of Operation.

3.3 Adaptation Layer

(R3) All FUNI implementations *Shall* support AAL5

All Modes of Operation

(O2) A FUNI implementation *May* support AAL3/4 concurrently with AAL5 on a per virtual connection basis

Mode 1b Only

3.4 FUNI Service Data Unit (SDU) Size

(R4) The FUNI *Shall* support up to 4096 octets of FUNI SDU for interfaces of smaller than or equal to DS1/E1 speeds

Modes 1a and 1b

(R5) The FUNI *Shall* support up to 9232 octets of FUNI SDU for interfaces of DS1/E1 speeds and above.

Modes 3 and 4

Note that there is no contradiction between requirements (R4) and (R5) above at the exact speeds of DS1 and E1. The equipment user may select Mode 1a or 1b for a 4096 octet FUNI SDU; or Mode 3 or 4 for a 9232 octet FUNI SDU.

(O3) FUNI SDUs up to 65535 octets may be supported. This option is consistent with the adaptation layer CPCS maximum length specified in ITU-T recommendation I.363.5.

All Modes of Operation

(CR1) When the "O3" option is implemented, the equipment implementor *Shall* declare the maximum FUNI SDU size supported by his equipment and the user of this option *Shall* notify the network operator of the SDU size used at time of subscription.

All Modes of Operation

Note that:

- For AAL5 connections, this entire FUNI SDU field is available for user information payload.
- For AAL3/4 connections, 8 octets of this field are dedicated to the AAL3/4 CPCS Header and Trailer and the remaining FUNI SDU octets are available for user information payload as shown in Figure 3.3. Therefore, the maximum size of the User Information Payload is 4088 and 9224 octets for the 4096 and 9232 octet frames respectively.

The term user information payload refers to all user information including upper layer encapsulations; it only excludes encapsulation related to AAL3/4 CPCS which is a part of the FUNI SDU as shown in Figure 3.1. In this diagram, the FUNI SDU is equal to the AAL3/4_CPCS_PDU as defined in ITU-T recommendation I.363.3.

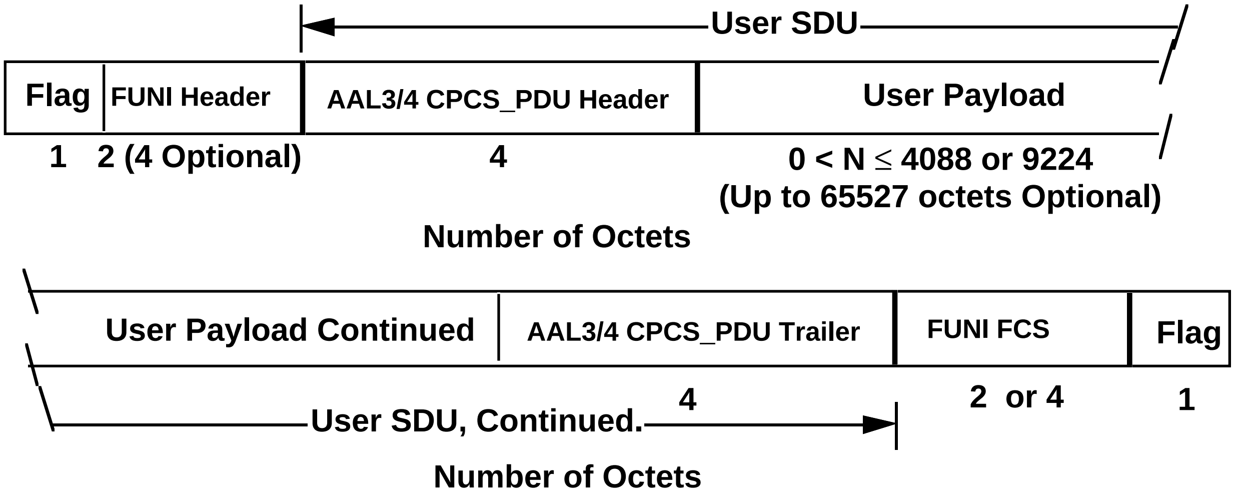


Figure 3.1 Illustration of FUNI SDU/AAL3/4_CPCS_PDU Encapsulation in a FUNI PDU

3.5 FCS Size

(R6) The FUNI *Shall* support a 16 bit CRC in the FCS field, per ITU-T recommendation Q.921, for interfaces of fractional speeds below DS1/E1

Modes 1a and 1b

(R7) The FUNI *Shall* support a 32 bit CRC in the FCS field, per ITU-T recommendation I.363 Addendum 1, Section 6.3.2.1.2: AAL5 CPCS Structure and Coding, for interfaces of DS1/E1 speeds and above.

Modes 3 and 4

3.6 Encapsulation/Decapsulation and SAR Requirements

On the User side of the FUNI:

(R8) Encapsulation/decapsulation of FUNI SDU in/off FUNI frames *Shall* follow the format shown in Figure 3.2 for AAL5 and Figure 3.3 for AAL3/4.

All Modes of Operation

On the Network side of the FUNI:

For AAL5

(R9) Encapsulation/decapsulation of the FUNI Frame Payload between FUNI Frames and AAL5_CPCS_PDU *Shall* follow the format shown in Figure 3.2.

All Modes of Operation

(R10) The SAR format *Shall* comply with AAL5 formats per ITU-T recommendation I.363.5 as illustrated in Figure 3.2. The SAR procedures *Shall* comply with AAL5 procedures per ITU-T recommendation I.363.5.

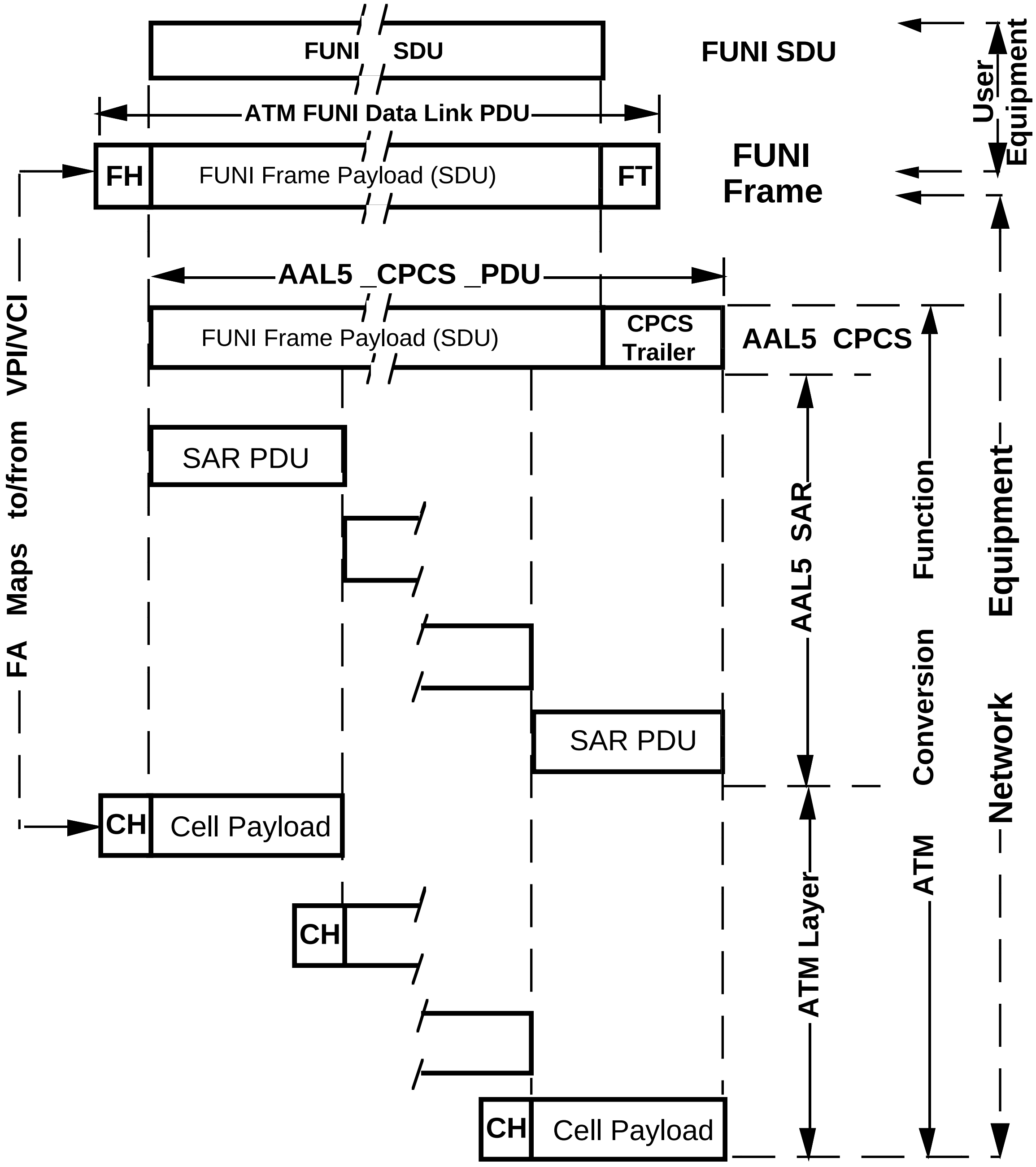
All Modes of Operation

For AAL3/4 (When AAL3/4 is implemented)

(CR2) Encapsulation/decapsulation of the FUNI Frame Payload between FUNI Frames and AAL3/4_CPCS_PDU *Shall* follow the format shown in Figure 3.3.

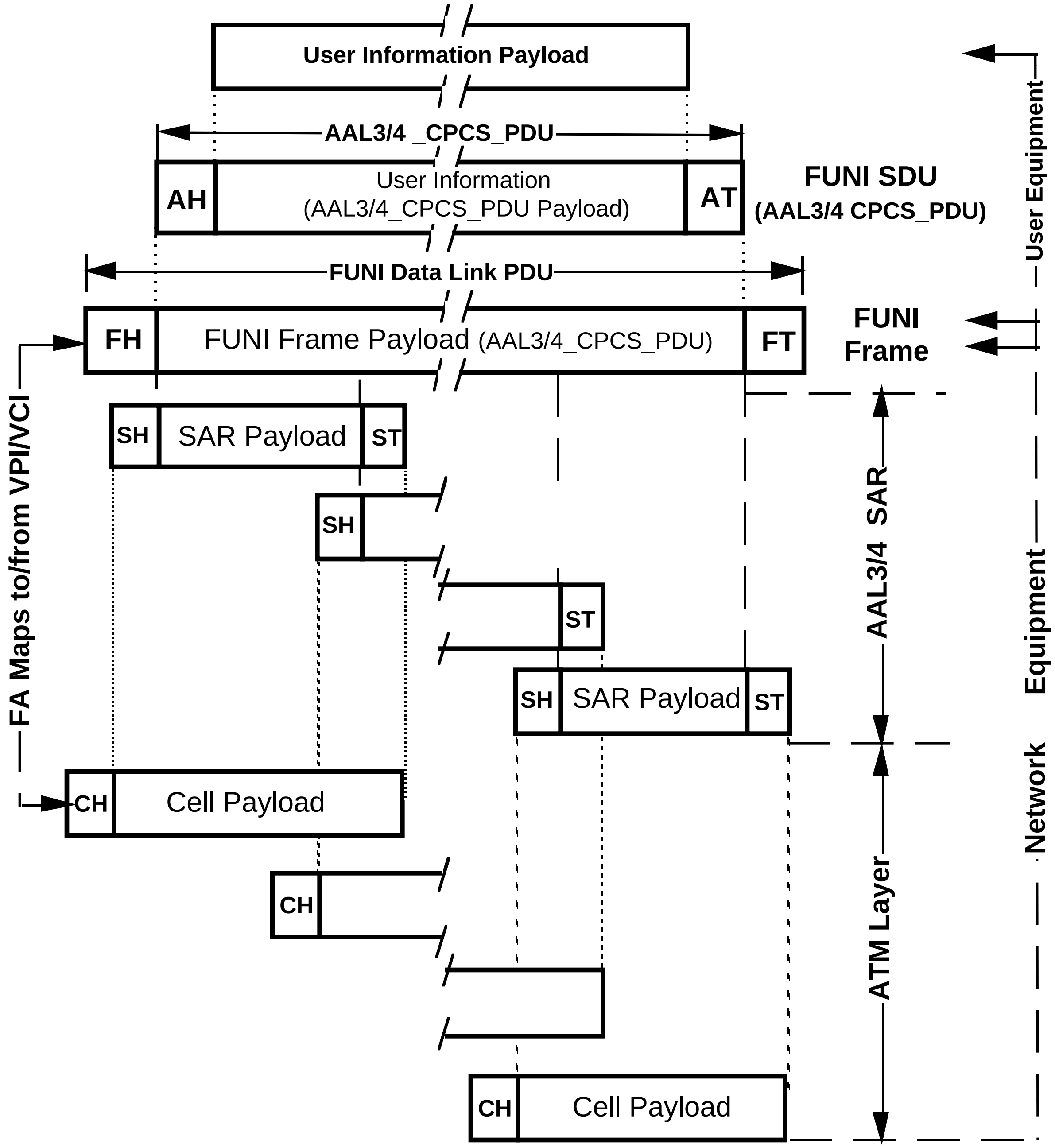
(CR3) The SAR format *Shall* comply with AAL3/4 formats per ITU-T recommendation I.363.3 as illustrated in Figure 3.3. The SAR procedures *Shall* comply with AAL3/4 procedures per ITU-T recommendation I.363.3.

Mode 1b Only For Both CR2 and CR3



FH = FUNI Frame Header
FT = FUNI/HDLC Frame Trailer = FCS
CPCS = Common Part Convergence Sublayer
CH = Cell Header
Note: HDLC Flags are not shown

Figure 3.2 Encapsulation/Decapsulation and SAR Formats of AAL5 FUNI
(Physical Layer is Not Shown in This Diagram)



FH = FUNI Frame Header	FT = FUNI/HDLC Frame Trailer	Note: HDLC Flags are not shown
AH = AAL3/4 CPCS Header	AT = AAL3/4 CPCS Trailer	
SH = SAR Header	ST = SAR Trailer	
CH = Cell Header		

Figure 3.3 Encapsulation/Decapsulation and SAR Formats of AAL3/4 FUNI (Physical Layer is Not Shown in This Diagram)

3.7 Number of Concurrent Reassembly Instances

(R11) A Compliant FUNI IWF *Shall* be capable of reassembling a minimum of 16 concurrent frames with interleaved arrival of their constituent cells. This requirement applies for both AAL5 and AAL3/4 implementations in Modes 1a and 1b only.

Modes 1a and 1b

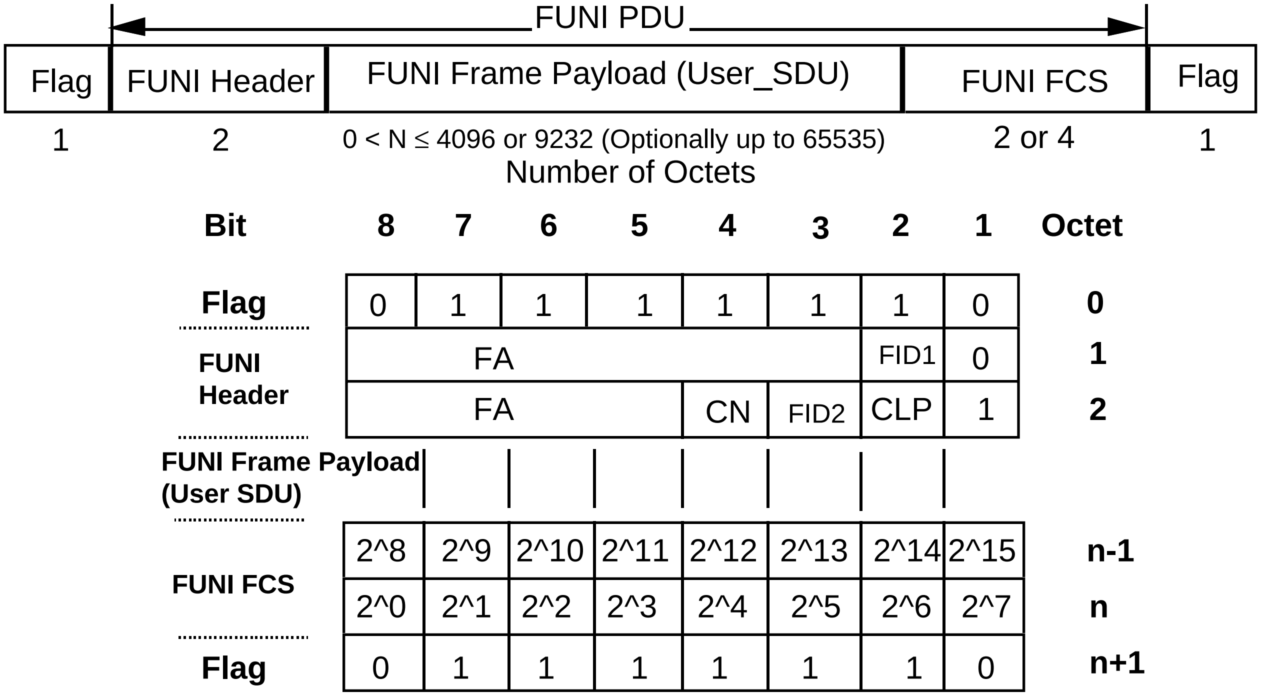
The split between the number of reassembly instances between AAL5 and AAL3/4 (in implementations which have both) is a matter of implementation and shall not be part of any requirements.

The above requirement was maintained in order to maintain compatibility between modes 1a and 1b in this FUNI v2.0 with the corresponding modes in the FUNI v1.0 [af-saa-0031.000] and the ATM DXI [af-dxi-0014.000] specifications. This requirement was dropped for the other FUNI v2.0 modes (3 and 4). Implementors should notice that a larger number of reassembly instances provides better frame loss and frame latency performance and consequently a better throughput.

3.8 Frame Format and Frame-to-Cell Header Mapping for Frames with the 2 Octet Header

(R12) The Frame structure across the FUNI *Shall* comply with Figure 3.4 for Frames with a two octet header.

Modes 1a, 1b and 3



Flag	FUNI Header	FUNI Frame Payload (User_SDU)	FUNI FCS	Flag
1	2	0 < N ≤ 4096 or 9232 (Optionally up to 65535)	2 or 4	1

Number of Octets

Bit	8	7	6	5	4	3	2	1	Octet
Flag	0	1	1	1	1	1	1	0	0
FUNI Header	FA						FID1	0	1
	FA				CN	FID2	CLP	1	2
FUNI Frame Payload (User SDU)									
FUNI FCS	2^8	2^9	2^10	2^11	2^12	2^13	2^14	2^15	n-1
	2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	n
Flag	0	1	1	1	1	1	1	0	n+1

Figure 3.4 FUNI Frame Format for the 2 octet Header Frames (shown with a 16 bit FCS)

(R13) The interpretation of the Frame Header fields *of the 2 octet frame header* and the mapping between the Frame Header and Cell Header fields *Shall* comply with the following definitions

FID *Frame Identification*: for identification of OAM cell vs. User Frames. These bits are set to 0 unless the frame is used for carrying an OAM cell. Specific values are defined in Table 3.2 below.

FA *Frame Address*: Octet 1, bit 6 through bit 3 are mapped to the 4 LSBs (least significant bits) of the VPI in the cell header. The four MSBs (most significant bits) of the VPI are not coded in the FA field, and are set to zero by the Network Equipment[1] on send and ignored on receive[2] Octet 1, bit 8 and bit 7, and Octet 2, bit 8 through bit 5, are mapped to the six LSBs of the VCI in the cell header. The ten MSBs of the VCI are not coded in the FA field, and are set to zero by the Network Equipment on send and ignored on receive. The all 0s FA is not mapped into the corresponding (all 0s) VPI/VCI which is a reserved value according to the UNI standards. Figure 3.5 shows the bit-to-bit mapping between the FA and VPI/VCI values.

FCS A 16 or 32 bit FCS (Frame Check Sequence) which complies with ITU-T recommendation Q.921 and I.363 requirements respectively (see R0).

CN If PTI = 01x in the last constituent ATM cell in a FUNI frame, then the Network Equipment sets CN (Congestion Notification) equal to one for that FUNI frame. Otherwise, the Network Equipment sets CN equal to zero. The User Equipment always sets the CN to zero.

CLP The Network Equipment copies the CLP Bit sent from the User Equipment into the CLP bit of all ATM cell headers constituting the FUNI frame. The CLP bit from the Network Equipment to the User Equipment is always set to zero.

Bit/Octet Order:

The left most bit of the octet (i.e., bit 8) is the Most Significant Bit (MSB). In diagrams with multiple octets the left most bit in the top-most octet is the MSB. These bits are transmitted right to left, top to bottom.

Modes 1a, 1b and 3

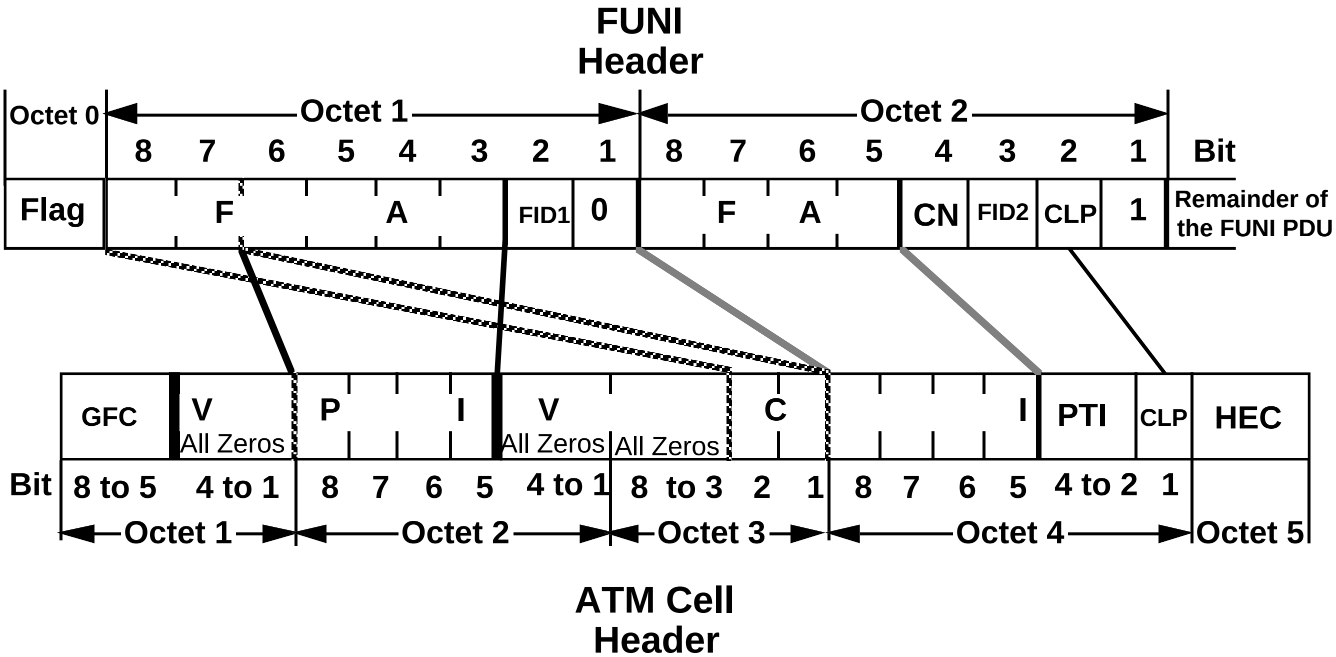
[1] This process is performed by the FUNI-to-ATM conversion function depicted in Figure 2.1 of this document

[2] Send and receive are in reference to the FUNI, i.e., sending is from the FUNI into the ATM network and receive is from the ATM network to the FUNI.

Table 3.2 below indicates the header's bit values of FID1 and FID2 associated with the corresponding types of frames.

Type of Frame	FID1	FID2
User Information (Data, Signaling or ILMI)	0	0
Reserved	0	1
OAM Cell	1	0
Reserved	1	1

Table 3.2 Frame Types Code Points



Note: Plain Text Indicates values in the Fields
Bold Text Indicates Names of the Fields and
Octet/Bit Order

Figure 3.5 Mapping Between the Frame Address (FA) and VPI/VCI Fields for Frames with 2 octet Header

The procedure of associating a particular virtual connection with a particular AAL is implementation specific and is beyond the scope of this document.

3.9 Frame Format and Frame-to-Cell Header Mapping for Frames with the 4 Octet Header

(CR4) The Frame structure across the FUNI *Shall* comply with Figure 3.6 for Frames with a four octet header.

Mode 4 Only

Bit	8	7	6	5	4	3	2	1	Octet
Flag	0	1	1	1	1	1	1	0	0
FUNI Header	FA (VPI, MSB)						FID1	0	1
	FA (VPI, LSB)		FA (VCI, MSB)		CN	FID2	CLP	0	2
	FA (VCI)							0	3
	FA (VCI, LSB)							1	4
FUNI Frame Payload (User SDU)									
FUNI FCS	2^24	2^25	2^26	2^27	2^28	2^29	2^30	2^31	n-3
	2^16	2^17	2^18	2^19	2^20	2^21	2^22	2^23	n-2
	2^8	2^9	2^10	2^11	2^12	2^13	2^14	2^15	n-1
	2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	n
Flag	0	1	1	1	1	1	1	0	n+1

Figure 3.6 FUNI Frame Format for the 4 octet Header Frames

(CR5) The interpretation of the Frame Header fields *of the 4 octet frame header* and the mapping between the Frame Header and Cell Header fields *Shall* comply with the following definitions. Mapping the FA to VPI/VCI fields are illustrated in Figure 3.7

FID *Frame Identification*: for identification of OAM cell vs. User Frames. These bits are set to 0 unless the frame is used for carrying an OAM cell. Specific values are defined in Table 3.2 above.

FA *Frame Address*: Octet 1, bit 8 through bit 3, and Octet 2, bit 8 and bit 7 represent the VPI. Octet 2, bit 6 and bit 5, and Octet 3, bit 8 through bit 2, and Octet 4, bit 8 through bit 2, represent the VCI. Figure 3.7 depicts this requirement.

FCS A 32 bit FCS (Frame Check Sequence) which complies with ITU-T recommendation Q.921 requirements.

CN If PTI = 01x in the last constituent ATM cell in a FUNI frame, then the Network Equipment sets CN (Congestion Notification) equal to one for that FUNI frame. Otherwise, the Network Equipment sets CN equal to zero. The User Equipment always sets the CN to zero.

CLP The Network Equipment copies the CLP Bit sent from the User Equipment into the CLP bit of all ATM cell headers constituting the FUNI frame. The CLP bit from the Network Equipment to the User Equipment is always set to zero.

Bit/Octet Order:

The left most bit of the octet (i.e., bit 8) is the Most Significant Bit (MSB). In diagrams with multiple octets the left most bit in the top-most octet is the MSB. These bits are transmitted right to left, top to bottom.

Mode 4 Only

FUNI 4 Octet Frame Header

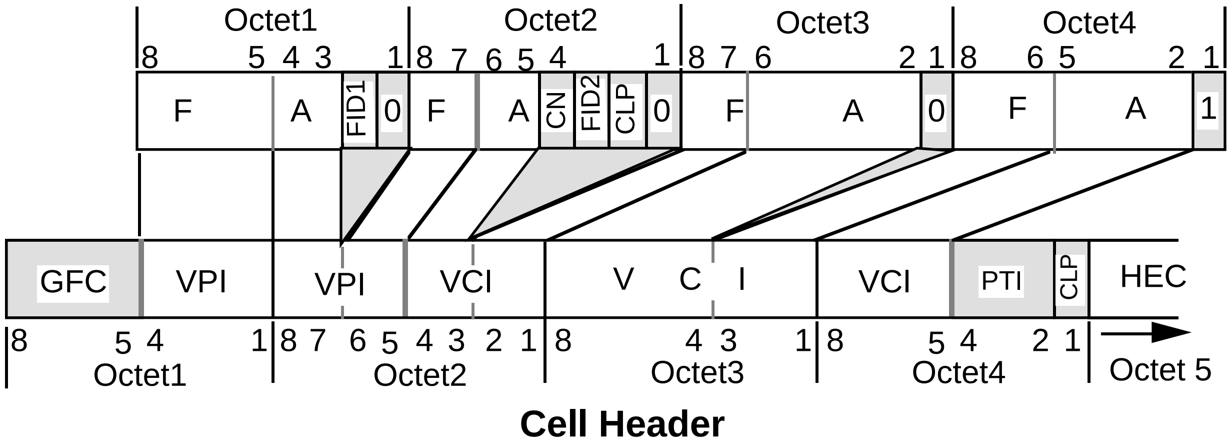


Figure 3.7 Mapping Between the Frame Address (FA) and VPI/VCI Fields for Four Octet Header Frames

4.0 PHYSICAL LAYER

There are two physical layers, explicitly specified here, that may be used in supporting this interface: DS1 or E1. Other physical layers may optionally be used at the implementor's discretion and responsibility as explained in Section 2 above.

When the DS1 physical layer is used it must meet the following requirement.

(R14) **The DS1** Physical Layer *Shall* meet all applicable requirements in the ANSI T1.403 and ANSI T1.408 specifications[3], [4] as stated in Sections 2.7.1 (Physical Medium Dependent Characteristic) and 2.7.2 (Transport Signal Format) of the ATM Forum DS1 Physical Layer Specifications, Version 1.0[5], with the exception of Section 2.7.1.2 (ATM Transfer Rate).

When the E1 physical layer is used, it has two potential framing modes: Structured and Unstructured. The **Structured** mode meets ITU-T's recommendation **G.704 framing** requirements and G.703 electrical interface requirements. The **Unstructured** mode meets only G.703 electrical interface requirements and *does not* meet G.704 framing requirements, i.e., it is **just a stream of serial bits**. Following are the requirements of the E1 physical layer.

(R15) **The E1 Structured** Physical Layer *Shall* be implemented and *Shall* meet all applicable requirements in the following ITU-T documents:

G.703-1991	Physical/Electrical Characteristics of Hierarchical Digital Interfaces
G.704-1991	Synchronous Frame Structures Used at Primary and Secondary Hierarchical Levels
G.709-1993	Synchronous Multiplexing Structure

(O4) In addition to the structured E1 physical layer, the Unstructured E1 Physical layer *May* be implemented.

(CR6) When Unstructured E1 physical layer is implemented, it *Shall* meet G.703 electrical interface requirements and the implementor *Shall* declare that his equipment supports this mode.

(CR7) Subscribers using the Unstructured E1 mode *Shall* notify network operator at time of subscription.

It should be emphasized that an Nx64 Kbps payload is carried as a single concatenated payload of N time slots in the DS1 or E1 frame. The time slots need not be contiguous. Supporting more than one Nx64 Kbps FUNI, or multiplexing with other types of interfaces, on one physical DS1/E1 interface is an optional implementation (represented in Figure 2.2b above) .

5. Frame-based UNI Service Definitions

5.1 Abstract Service Model

An abstract service model is used to facilitate understanding the services exchanged between the different layers within an interface. In particular, it shows how various protocol parameters are used and how service data units (SDUs) and protocol data units (PDUs) relate to each other. It also provides a framework which facilitates a comprehensive analysis of the inter-layer interfaces.

The abstract service model is stated in terms of primitive functions and the parameters of those primitives. For the purpose of this specification, the model addresses the services provided by the FUNI Data Link Layer and the AAL5 Common Part Convergence Sublayer (AAL5-CPCS).

The AAL5-CPCS service is specified in the ITU-T recommendation I.363 Section 6 (contained in I.363 addendum 1).

The FUNI Data Link Layer service is defined in this section.

5.2 Application of the FUNI Data Link Layer (FUNI-DL) Service

Upper layer user applications do not directly use the FUNI Data Link (FUNI-DL) service primitives, but rather use the AAL5-CPCS service (with the exception of the ATM layer functions on the user-side when the optional OAM cell transfer capability is implemented; in which case the FUNI-DL service primitives may be directly used).

The FUNI-DL service is exchanged with the AAL5-CPCS service at the interworking function provided by the FUNI-DL network-side. Thus, unlike the AAL5-CPCS service, the FUNI-DL service is *not* an end-to-end service, it only spans the segment of the connection between the FUNI-DL user-side and FUNI-DL network-side.

A draft ITU-T recommendation, Q.2119 “B-ISDN ATM Adaptation Layer-Convergence Function for SSCOP Above the Frame Relay Core Service,” can be slightly modified and used as described below to interchange AAL5-CPCS service primitives with FUNI-DL service primitives.

Applications written to the AAL5-CPCS service definition (using message mode only and without using the corrupted data delivery option) can be run directly over AAL5-CPCS or over a combination of the modified ITU-T Q.2119 and FUNI-DL. This includes any usual AAL5-CPCS service user such as UNI 3.1 SSCOP and associated signaling stacks, ILMI, etc.

Figure 5.1 below illustrates the relationship among the various layers.

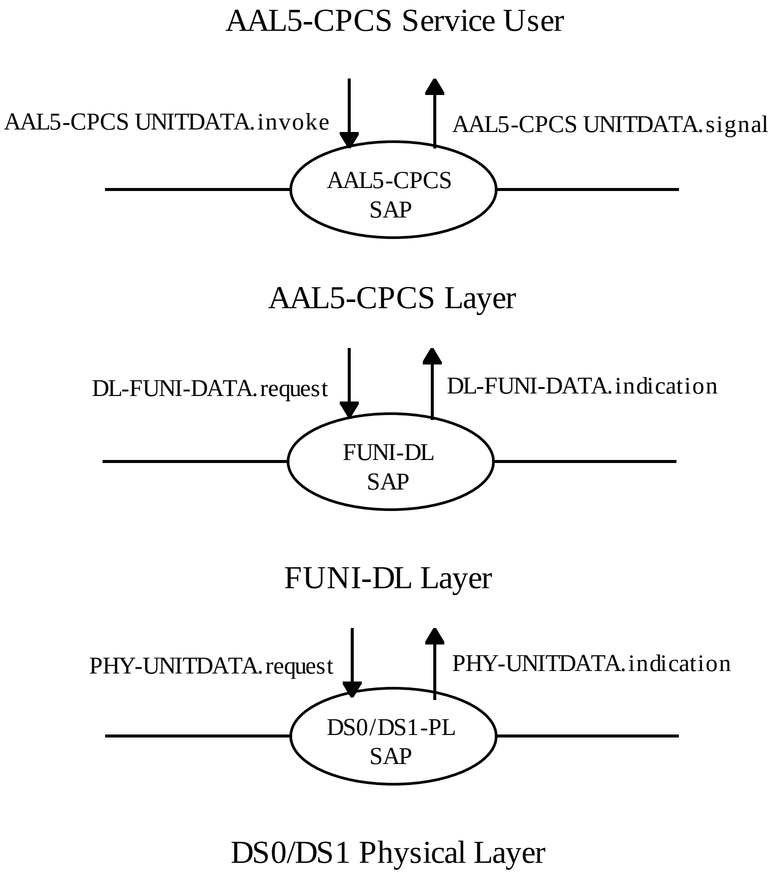


Figure 5.1: FUNI-DL Service Inter-Layer Relationships

5.3 FUNI DATA LINK LAYER SERVICE DEFINITION

The FUNI-DL layer service provides the primitives and parameters shown in Table 5.1.

(R16) The FUNI *Shall* use AAL5 in message mode *Only* and *Shall Not* use the corrupted data delivery option.

(R17) The FUNI *Shall* not support the transfer of the AAL5-CPCS-User to User parameter.

The following Table 5.1 states the required and optional Data Link Layer primitives. An "X" means a required parameter and an "O" means an optional parameter which is implemented only if the OAM frames option is implemented.

	DL-FUNI-DATA.request	DL-FUNI-DATA.indication
FUNI-user-data	X	X
FUNI-CLP	X	X
FUNI-EFCI	X	X
FUNI-OAM	O	O

Table 5.1: FUNI Data Link Layer Service Primitives and Parameters

The following two sections, 5.3.1 and 5.3.2 are Normative definitions used in conjunction with Table 5.1.

5.3.1 The DL-FUNI-DATA.request Primitive

The DL-FUNI-DATA.request primitive is used by the FUNI-DL service user (the user-side protocol stack or the network-side interworking function) to access the FUNI Service Access Point (FUNI SAP) in order to pass a Service Data Unit (SDU) and other service parameters to the FUNI-DL Layer.

When invoking a DS-FUNI-DATA.request, the parameters that are passed include:

- FUNI-user-data: the SDU passed via this parameter contains AAL5-CPCS Interface Data if the FUNI-OAM parameter = 0 and contains an OAM cell if the FUNI-OAM parameter = 1.
- FUNI-CLP: carries the AAL5-CPCS-Loss Priority signal.
- FUNI-EFCI: carries the AAL5-CPCS-Congestion Indication signal.
- FUNI-OAM: used to indicate that the contents of the FUNI-user-data parameter is an OAM cell. This parameter is set to 0 if the FUNI-user-data parameter is to be interpreted as user data (i.e., AAL5-CPCS Interface Data), or set to 1 if the FUNI-user-data parameter is to be interpreted as an OAM cell. More specifically, this parameter is optional, but is implemented whenever the optional OAM cell support is implemented.

1. DL-FUNI-Data.request: if FUNI-OAM is set to 0, then Bit 2 of Octet 1 and Bit 3 of Octet 2, in the FUNI-DL frame header, are set to 0; else Bit 2 of Octet 1 is set to 1 and Bit 3 of Octet 2 is set to 0.
2. DL-FUNI-Data.indication: if Bit 2 of Octet 1, in the FUNI DL Frame Header, is set to 1 and Bit 3 of Octet 2 is set to 0, then FUNI-OAM is set to 1; else it is set to 0.

5.3.2 The DL-FUNI-DATA.indication Primitive

The DL-FUNI-DATA.indication primitive is received by the FUNI-DL service user (the modified Q.2119 sublayer or the network-side interworking function) at the FUNI Service Access Point (FUNI SAP) to allow the FUNI-DL layer to pass a Service Data Unit (SDU) and other service parameters to the FUNI-DL service user (the modified Q.2119 sublayer or the network-side interworking function).

When a DS-FUNI-DATA.indication is received, the parameters that are passed include:

- FUNI-user-data: the SDU passed via this parameter contains AAL5-CPCS Interface Data if the FUNI-OAM parameter = 0 and contains an OAM cell if the FUNI-OAM parameter = 1.
- FUNI-CLP: carries the AAL5-CPCS-Loss Priority signal.
- FUNI-EFCI: carries the AAL5-CPCS-Congestion Indication signal.
- FUNI-OAM: indicates the type of payload in the FUNI-user-data parameter; if equal to 0, the FUNI-user-data parameter is to be interpreted as AAL5-CPCS Interface Data, or if set to 1, the FUNI-user-data parameter is to be interpreted as an OAM cell.

5.4 THE CPCS SERVICE OF AAL5

This section is Information only. The Common Part Convergence Sublayer (CPCS) of AAL5 is defined in ITU-T Recommendation I.363 Section 6. Both the signaling and ILMI functions use only message mode as defined for AAL5 and neither use the “corrupted data delivery” option. Table 5.2 summarizes the primitives and parameters of the AAL5 CPCS service.

	CPCS-UNITDATA.invoke	CPCS-UNITDATA.signal
Interface Data (ID)	X	X
More (M) (Note 1)	-	-
CPCS-Loss Priority (CPCS-LP)	X	X
CPCS-Congestion Indication (CPCS-CI)	X	X
CPCS-User-to-User-Indication (CPCS-UU) (Note 3)	X	X
Reception Status (RS) (Note 2)	-	-

Note 1: Not present in message mode.
Note 2: Not present as the corrupted data delivery option is not used.
Note 3: FUNI ignores/sets to 0 the AAL5-CPCS-UU parameter.

Table 5.2: AAL5 CPCS Service Primitives and Parameters

6. Mapping Between FUNI and AAL5 Primitives

When used with FUNI, the Q.2119/FUNI convergence layer provides a mapping of service requests between the FUNI service and the AAL5-CPCS service. The Q.2119/FUNI convergence layer, a variant of the ITU-T Q.2119 convergence layer, is defined as follows.

6.1 PDU Format

As shown in Figure 6.1, Q.2119/FUNI-user data SDUs are embedded in the FUNI data link frame immediately following the FUNI frame header (Octets 3 to n). Some applications (such as SSCOP) require that they receive SDUs aligned on a 32 bit boundary. Therefore, when received, the FUNI data link frame should be aligned so that octet 3 is on a byte address that is 0 mod 4 or the SDU must be copied to align on a 32-bit boundary before the SDU is passed to the upper layer.

Octets

1 and 2	3 to n	n+1 to n+2
Header	FUNI-DL-DATA SDU (AAL5 CPCS Interface Data or OAM Cell)	FCS

Figure 6.1: FUNI Data Link Frame

6.2 MAPPING THE AAL5 CPCS INVOKE INTO THE FUNI SERVICE PRIMITIVES

As shown in Figure 6.2, upon receipt of a CPCS-UNITDATA.invoke signal from the Q.2119/FUNI-user, a DL-FUNI-DATA.request primitive is formed with the parameters noted below and submitted to the FUNI service:

1. The FUNI-user-data parameter is formed as specified in Section 5.3.1 using the “interface data” parameter of the CPCS-UNITDATA.invoke signal (e.g., SSCOP PDU, ILMI PDU, other user data).
2. The “CLP” parameter is formed using the “CPCS-LP” parameter of the CPCS-UNITDATA.invoke signal (and which will be “0” for signaling messages).
3. The “EFCI” parameter is set to “0”.

The CPCS-Congestion Indication and CPCS-User-to-User signals are ignored.

Note that each instance of the protocol stack is associated with a particular connection and so “knows” the appropriate Frame Address to use (i.e., the addressing information is not passed via the CPCS service primitive).

6.3 MAPPING THE FUNI SERVICE PRIMITIVES INTO THE AAL5 CPCS SIGNAL

As shown in Figure 6.2, upon receipt of a DL-FUNI-DATA.indication primitive from the FUNI service, a CPCS-UNITDATA.signal signal is formed with the parameters noted below and sent to the CPCS user (e.g., SSCOP, ILMI, other application):

1. The “Interface Data” parameter is set to the value in the DL-FUNI-Data SDU field of the “FUNI-user-data” parameter of the DL-FUNI-DATA.indication primitive (see Section 6.1 and Table 5.1).
2. The “CPCS-LP” parameter is set to the value in the “FUNI-CLP” parameter of the DL-FUNI-DATA.indication primitive.
3. The “CPCS-CI” parameter is set to the value in the “FUNI-EFCI” parameter of the DL-FUNI-DATA.indication primitive.
4. The “CPCS-UU” parameter is set to zero.

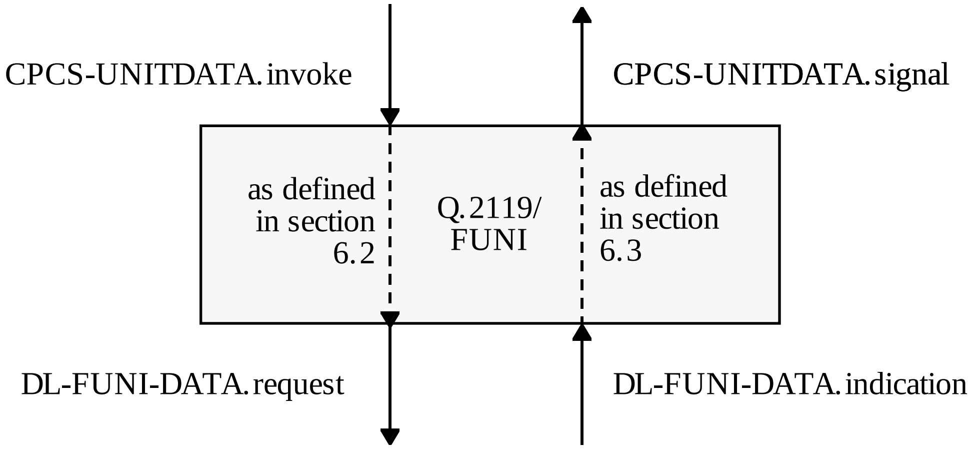


Figure 6.2: Relationship of AAL5-CPCS Signals and FUNI Service Primitives

6.4 USER-SIDE (DTE) BEHAVIOR

Transmitter: the CPCS user (e.g., SSCOP, ILMI) shall use the CPCS-UNITDATA.invoke primitive with the user data, e.g., SSCOP PDU, passed as the Interface Data parameter, with the CPCS-LP parameter set to "0", and with the CPCS-CI parameter set to "0". The Q.2119/FUNI convergence function shall pass this information to the FUNI service as defined in Section 6.2.

Receiver: a FUNI-DATA.indication shall signal the arrival of a Frame, e.g., a signaling message. The Q.2119/FUNI convergence function shall map the signals as defined in Section 6.3 into AAL5-CPCS signals to deliver the DL-FUNI-Data SDU to the CPCS user. The FUNI-CLP and FUNI-EFCI parameters shall be ignored.

6.5 NETWORK-SIDE (DCE) BEHAVIOR

6.5.1 Network to User Translation

Translation of received AAL5 CPCS PDUs into FUNI Data Link PDUs shall use the same mechanism described in Section 6.2 except that the FUNI-EFCI parameter is formed from the CPCS-CI parameter, rather than being set to "0".

6.5.2 User to Network Translation

Translation of received FUNI Data Link PDUs into AAL5 CPCS PDUs shall use the same mechanism described in Section 6.3.

6.6 LAYER MANAGEMENT

There are no interactions with layer management defined. Interactions to report the values of the congestion management parameters of the FUNI service are not necessary. All such interactions may be performed by the CPCS user or the OAM cell management entity.

7.0 SPECIFICATION OF SIGNALING OVER FUNI

(R18) VPI/VCI values for the signaling channel *Shall* be assigned consistent with the UNI 4.0 specifications (e.g., default to VPI=0, VCI=5).

7.1 SIGNALING RESTRICTIONS AND DEFAULTS

All the following parts of this section, Section 7.1, are Normative

The FUNI is capable of supporting VBR and UBR traffic classes and AAL5 and AAL3/4 only. This document defines action upon receiving other legal UNI requests across a FUNI.

7.1.1 ATM Adaptation Layer

A connection setup request may only request AAL type 5 and AAL type 3/4 in the ATM Adaptation Layer Information Element. Requests for AAL type 1 and User-defined AAL must be rejected with Cause Code #78 (“AAL parameters cannot be supported”) for UNI 3.1 environment and Cause Code #93 (same cause) in SIG 4.0 environment.

(R19) If a connection setup request is received with an ATM Adaptation Layer Information Element (AAL IE) specifying AAL5, the FUNI interworking function *Shall* use AAL5 regardless of the Mode in which the interface is configured.

(R20) If a connection setup request is received with an ATM Adaptation Layer Information Element (AAL IE) specifying AAL3/4, and the FUNI interface is configured in Mode 1b, the FUNI interworking function *Shall* use AAL3/4. Otherwise, the connection *Shall* be rejected with Cause Code #78 (“AAL parameters cannot be supported”) in a UNI 3.1 environment. Cause Code #93 (“AAL parameters cannot be supported”) *Shall* be used in a SIG 4.0 environment.

(R21) If a connection setup request is received without an ATM Adaptation Layer Information Element (AAL IE), the FUNI interworking function *Shall* use AAL5.

7.1.2 Broadband Bearer Class

The FUNI is capable of supporting only Broadband Bearer Classes C and X, since these are the only bearer classes defined in UNI 3.1 and UNI 4.0 that support variable bit rate connections. Connection requests specifying Bearer Classes C and X must otherwise comply with UNI 3.1 or UNI 4.0 (e.g., combinations with Quality of Service parameters and ATM Adaptation Layer parameters).

Broadband Bearer Class X requires that an end-to-end cell relay connection between users be established. Although FUNI requires a frame-to-cell conversion in the network with attendant processing of AAL5 or AAL3/4, class X is allowed in order to support existing applications such as those using RFC 1577.

Broadband Bearer Class A implies timing is required and Usage Parameter Control will police at Peak Cell Rate, which is not consistent with the variable bit rate, bursty nature of FUNI traffic. Therefore, connection setup requests specifying Bearer Class A must be rejected with Cause Code #65 (“bearer capability not implemented”).

7.1.3 Quality of Service Parameter

The FUNI is only capable of supporting QoS classes 0 (unspecified) and 3. Although the categorization of QoS class 3 as “variable bit rate” is not standardized, this interpretation is generally accepted. A connection setup request containing a request for QoS classes 1 and 4 must be rejected with Cause Code #49 (“Quality of Service unavailable”).

7.1.4 Traffic Management

Non-real time Variable Bit Rate (nrt-VBR) and Unspecified Bit Rate (UBR) are the only traffic classes supportable by the FUNI. Constant Bit Rate (CBR), real time Variable Bit Rate (rt-VBR), and Available Bit Rate (ABR) cannot be supported; connection setup requests specifying these traffic classes must be rejected with Cause Code #73 (“unsupported combination of traffic parameters”).

7.2 SIGNALING PROTOCOL STACK

The FUNI is capable of supporting signaling by running the UNI signaling protocol stack elements of UNI/Q.2931, SSCF, SSCOP/Q.2110 over the FUNI data link layer connection using AAL5 in message mode without using the corrupted data delivery option.

Once at the DCE interworking function, the FUNI Data Link framing will be replaced by AAL5/I.363 and the Signaling PDU will then be properly formatted for delivery to the Signal Processor. It does not matter whether the UNI is UNI 3.1 or 4.0 at the FUNI since the SSCOP layer for UNI 3.1 and 4.0 are the same. [Note: this specification does not address support for UNI 3.0 signaling which uses a different version of the SSCOP].

The FUNI user-side is required to support the full UNI 3.1/4.0 signaling stack including messages and procedures that correspond to the signaling entity controlling the FUNI port on the network-side. The layers of the signaling stack that are occupied by AAL5 and ATM in the UNI 3.1 and 4.0 specifications are replaced by a convergence function (to be defined here) and the FUNI data link layer on the FUNI user-side. No additional functionality is required of the FUNI network-side.

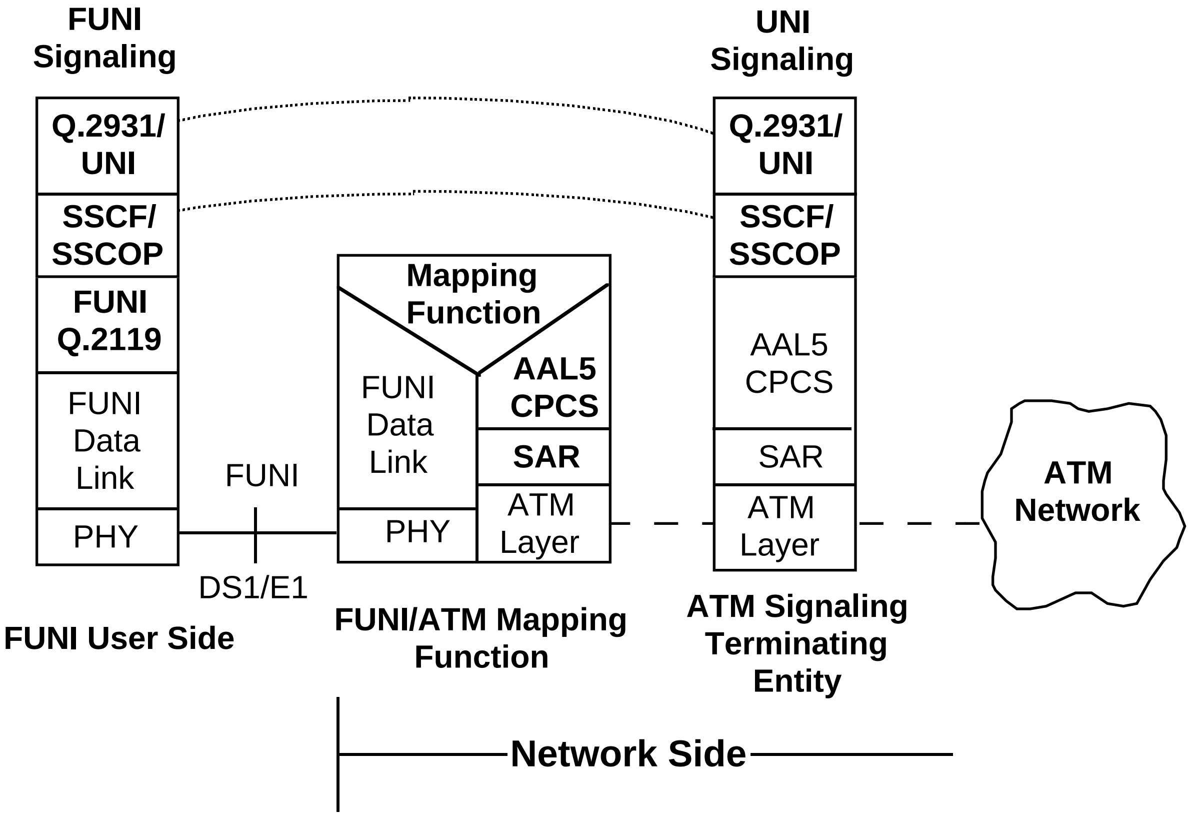


Figure 7.1: Signaling over FUNI

8.0 Specification of ILMI over FUNI

(R22) The ILMI Protocol version used in a specific FUNI implementation *Shall* be consistent with the Signaling Protocol version used in the same implementation

(R23) ILMI messages *Shall* be transmitted across the FUNI by placing the SNMP message which would have been carried in an AAL5_CPCS_PDU in a FUNI Data Link frame on the FUNI user-side. The connection *Shall* use a default permanent channel address, VPI/VCI, of 0/16; and *Shall* be configured to use AAL5 in message mode without use of the corrupted data delivery option. The method of establishing the ILMI channel is beyond the scope of this specification (e.g., provisioning).

(R24) The SNMP messages on the user-side *Shall* be encapsulated directly in the DL-FUNI-DATA field without the use of LLC/SNAP or other multiprotocol encapsulation.

8.1 ILMI Use of CPCS Service Primitives

ILMI does not explicitly make use of the CPCS Service Primitives in the various ILMI specifications (3.1 and 4.0), but it does make use of the services provided by the AAL5-CPCS.

(R25) The functions of the ILMI specification *Shall* be mapped onto AAL5-CPCS service primitives in the Interworking function as illustrated by Figure 8.1.

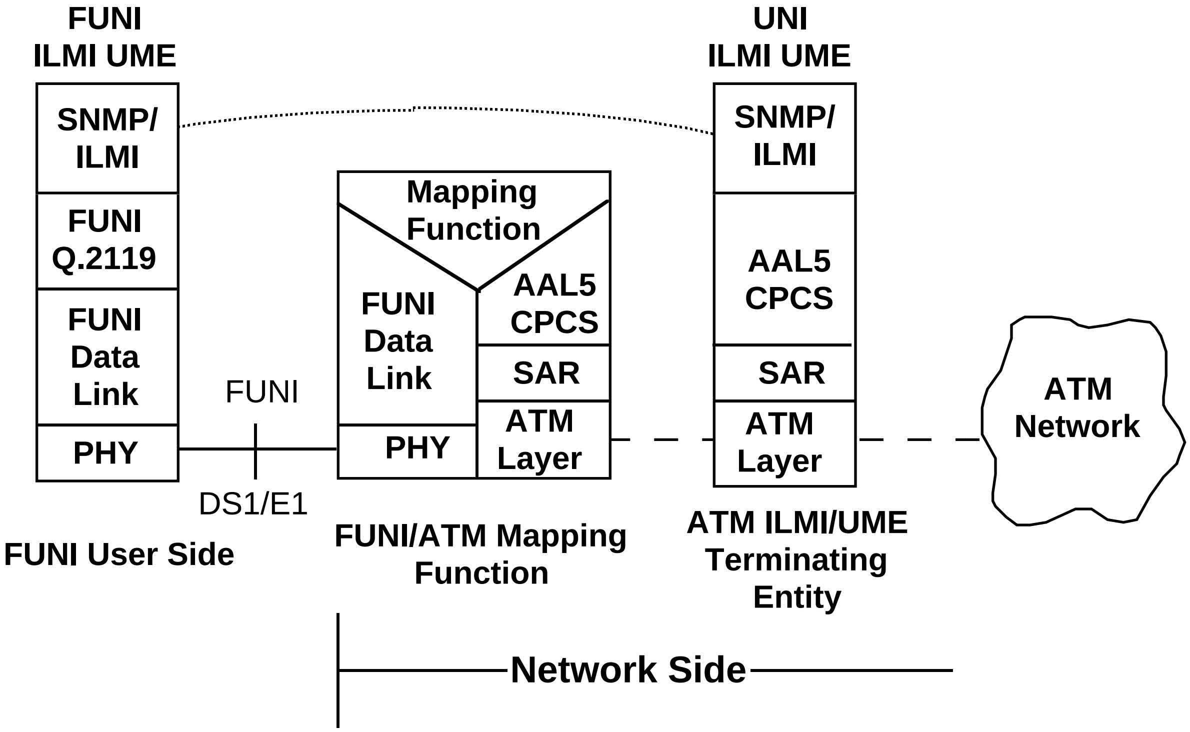


Figure 8.1: ILMI over FUNI

The following behaviors and formats are *Normative*.

8.2 User-Side (DTE) Behavior

Transmitter: SNMP ILMI shall use the CPCS-UNITDATA.invoke primitive with the ILMI SNMP PDU passed as the Interface Data parameter, with the CPCS-LP parameter set to "0", and with the CPCS-CI parameter set to "0". The Q.2119/FUNI convergence function shall pass this information to the FUNI service as defined in Section 6.2.

Receiver: a FUNI-DATA.indication shall signal the arrival of an ILMI message. The Q.2119/FUNI convergence function shall map the signals as defined in Section 6.3 into AAL5-CPCS signals to deliver the ILMI SNMP PDU to the ILMI layer. The FUNI-CLP and FUNI-EFCI parameters shall be ignored.

8.3 NETWORK-SIDE (DCE) BEHAVIOR

8.3.1 Network to User Translation

Translation of received AAL5 CPCS PDUs into FUNI Data Link PDUs shall use the same mechanism described in Section 6.2 except that the FUNI-EFCI parameter is formed from the CPCS-CI parameter, rather than being set to "0".

8.3.2 User to Network Translation

Translation of received FUNI Data Link PDUs into AAL5 CPCS PDUs shall use the same mechanism described in Section 6.3.

8.4 PDU FORMAT

As shown in Figure 8.2 ILMI SNMP messages shall be embedded in the FUNI data link frame immediately following the FUNI frame header. So that the ILMI (and other SDUs received by the FUNI layer) operates the same regardless of upper layer service user (e.g., SSCOP), the DL-FUNI-DATA-PDU should be aligned so that octet 1 of the FUNI PDU is on a byte address that is 2 mod 4 or the SDU must be copied to align on a 32-bit boundary before the SDU is passed to the upper layer.

Octets

1-2 or 1 to 4	3 to n or 5 to n	n+1 to n+2 or n+1 to n+4
Header	ILMI SNMP PDU	FCS

Figure 8.2: PDU Format to Embed an ILMI SNMP PDU in a FUNI Data Link Frame

8.5 LAYER MANAGEMENT

There are no interactions with layer management defined. Interactions to report the values of the congestion management parameters of the FUNI service are not necessary. The upper layers (AAL5-CPCS and ILMI SNMP) should perform all such interactions.

9.0 TRAFFIC MANAGEMENT IN THE FUNI-TO-ATM INTERWORKING FUNCTION

9.1 REFERENCE CONFIGURATION

Figure 9.1 depicts a reference configuration for the traffic management function across the FUNI interface and in the FUNI-to-ATM Interworking Function. Points X and Y on the diagram designate specific points where the actual traffic characteristics can be measured according to conformance specifications on either side of the demarcation point (X or Y). Depending on the implementation of the FUNI interworking function and where it is physically located in the network element containing it, points A and B may not be open interface points accessible for measurements of conformance. The functional and conformance requirements below take the preceding fact into consideration. The following requirements and specifications utilize the reference configuration in Figure 9.1.

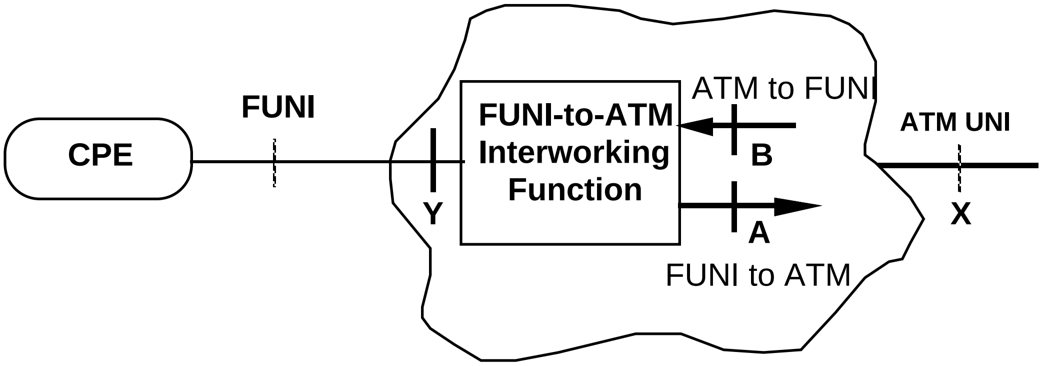


Figure 9.1 Reference Configuration for Traffic Management Specifications of the FUNI

9.2 TRAFFIC SHAPING

It is recognized that a traffic shaping function in the user equipment (CPE) and/or the FUNI interworking function will help increase the network efficiency and throughput of the connections supported on the FUNI interface. Traffic shaping refers to controlling and/or modifying the traffic characteristics, i.e., cell transmission rate, cell emission spacing and burst sizing, to assure its conformance to a contracted connection parameters policed by a UPC function. This shaping function is most important in the direction of Frame to ATM shown in Figure 9.1. On the basis of the importance of this function, the following set of requirements are recommended.

9.2.1 CPE Traffic Shaping

There is No traffic shaping requirement on traffic emitted from the FUNI IWF in the direction of Network to CPE (ATM to Frame).

At Point of Demarcation Y**,** Referring to Figure 9.1:

(O5) The user equipment (CPE) *May* at its option implement a traffic shaping function performed on the frames emitted on the FUNI in the CPE to Network (Frame to ATM) direction.

If the CPE implements the traffic shaping function stated in the option above, the implementation shall meet the following conditional requirements.

(CR8) For contracted parameters PCR, SCR and MBS[3] at the ATM UNI interface (Frame to ATM direction at point X on Figure 9.1): the CPE *Shall* control (shape) the average traffic rate to SCR.

(CR9) It *Shall* be possible to enable/disable the shaping option by the user of the equipment.

The following should be noted. If the CPE does not implement the traffic shaping function: the frame traffic may overflow the IWF's shaping buffer - and result in frame loss at the IWF - whenever the frame traffic from the CPE violates the traffic contract.

Furthermore, even if the optional CPE traffic shaping is implemented and the average traffic rate is below SCR over an integrated time interval, a long burst of frames within that interval may still overflow the IWF's shaping buffer.

[3] These parameters have the standard GCRA meaning, i.e., Peak Cell Rate, Sustainable Cell Rate and Maximum Burst Size.

9.2.2 FUNI Interworking Function Traffic Shaping

There is no requirement for traffic shaping in the FUNI IWF in the ATM to frame direction (traffic received across point B from the next ATM layer in the network element to the ATM side of the FUNI IWF).

Referring to Figure 9.1:

(R26) The FUNI IWF *Shall* implement a traffic shaping function in the Frame to ATM direction (traffic transmitted across point A from the ATM side of the FUNI IWF to the next ATM layer in the network element, see Figure 2.1 for protocol stack).

(R27) The Traffic Shaping Function in the FUNI IWF *Shall* control the traffic characteristics (cell emission and burst rates) to conform with the following formulas, which are illustrated in figure 9.2:

The maximum cell emission rate *Shall* not exceed 1/PCR with a Limit Lp (maximum cumulative clumping time) that does not exceed the contracted CDVT. This assures conformance to a Peak Cell Rate GCRA (1/PCR, CDVT)

The FUNI IWF may emit cell bursts at the PCR. The number of cells per burst *Shall* not exceed MBS and *Shall* have a clumping Limit Ls. Where,

Ls = (MBS-1) (1/SCR - 1/PCR) + CDVT.

This conforms to an Sustainable Cell Rate GCRA (1/SCR, Ls) and a Maximum Burst Size (MBS).

To assure no loss of cells, the period Ts between maximum sized bursts *Shall* conform to the formula Ts = Ls + 1/SCR

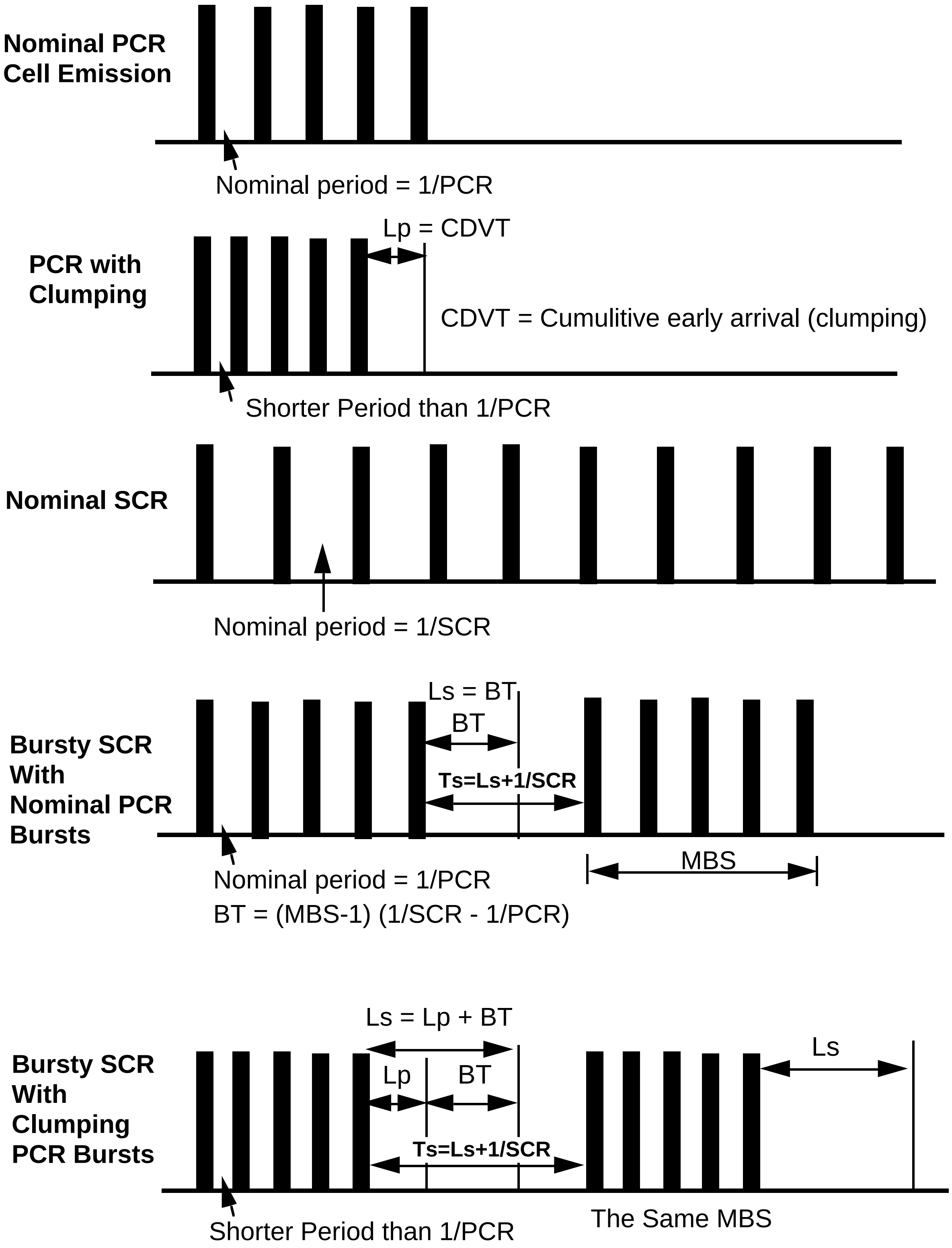

Nominal PCR Cell Emission
Nominal period = 1/PCR
PCR with Clumping
Lp = CDVT
CDVT = Cumulitive early arrival (clumping)
Shorter Period than 1/PCR
Nominal SCR
Nominal period = 1/SCR
Bursty SCR With Nominal PCR Bursts
Ls = BT
BT
Ts=Ls+1/SCR
MBS
Nominal period = 1/PCR
BT = (MBS-1) (1/SCR - 1/PCR)
Bursty SCR With Clumping PCR Bursts
Ls = Lp + BT
Lp
BT
Ls
Ts=Ls+1/SCR
Shorter Period than 1/PCR
The Same MBS

Figure 9.2 Illustration of Traffic Shaping Requirement in the FUNI IWF

9.3 TRAFFIC CONFORMANCE

There is no requirement of policing function in the ATM to Frame direction at the demarcation point X of Figure 9.1.

There is no requirements of policing in the frame domain at demarcation point Y in either direction of transmission.

Referring to Figure 9.1,

(R28) The ATM traffic crossing demarcation point X in Figure 9.1, in the Frame to ATM direction, *Shall* conform to the standard UNI 3.1 (Section 3.6.2) traffic contract, and UNI signaling 4.0, Annex 9[4], with the following parameters: PCR, SCR, MBS and CDVT. These parameters *Shall* be set by agreement between the user and the network provider operating the FUNI IWF. How the agreement is set is beyond the scope of this document.

10.0 HANDLING OF OAM CELLS IN FUNI

This FUNI specification allows the optional support of Frame-to-OAM cells Interworking. This section addresses how the OAM cells are handled.

(O6) Support of OAM Frame-to-OAM Cell Interworking on the FUNI is optional

(CR10) If the preceding option is supported, it *Shall* be possible to enable or disable this function per FUNI interface

(CR11) For a given VCC: When an OAM cell arrives from the network to the ATM side of the FUNI in the middle of a cell stream being reassembled for frame transmission on the FUNI to the user, the OAM cell *Shall* be framed and transmitted to the user in the first available opportunity regardless of the time of completing the reassembly of the stream it came with.

(CR12) Only F5 segment and end-to-end OAM cells are supported.

Note that F4 cannot be supported because there is no VP service supported and physical layer OAM cells are inappropriate on a FUNI

[4] "Guidelines on the use of Bearer Class, Traffic Parameters and QoS."

10.1 MAPPING OAM CELLS INTO FUNI DL FRAMES

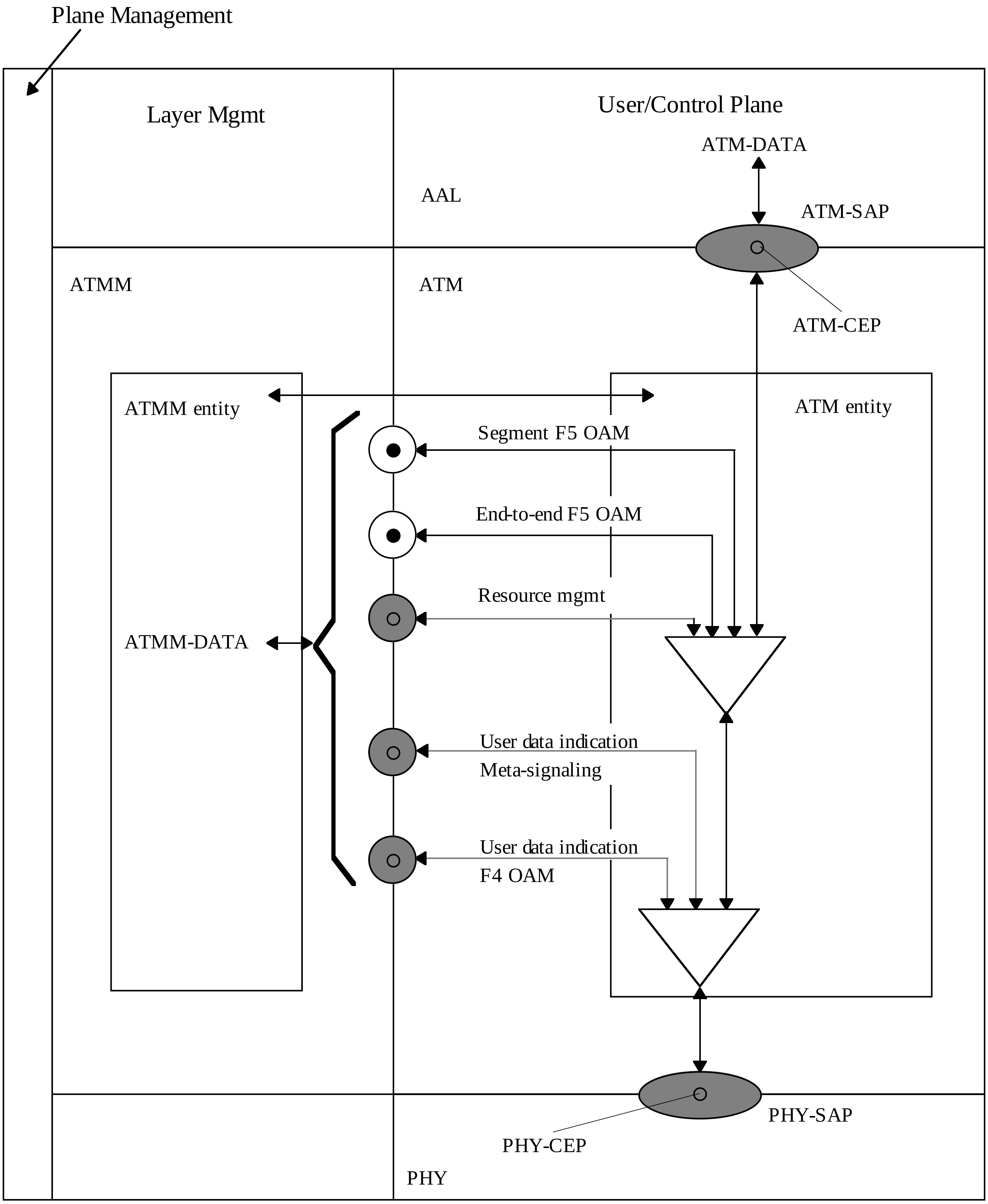
10.1.1 Objectives and Overview

OAM cells are passed to the FUNI layer (and back) using the FUNI Service Primitives defined in earlier sections of this document. The mapping of service primitives and parameters between the ATM Management Plane and FUNI Data Link is defined here. Note that this is done without using the modified Q.2119/FUNI, which defines a mapping to be used by an AAL5 CPCS user carried over FUNI service. To properly accomplish this function, the two PTI codes relating to OAM cells shall be carried with the OAM cell to allow the FUNI user to distinguish segment flows from end-to-end flows. Carrying the Payload Type Indication is accomplished by carrying a full 53 octet ATM OAM cell within the FUNI DL frame. In the network-to-user direction, the received OAM cell is mapped into the FUNI-Data Link frame User-SDU field in its entirety. In the user-to-network direction (e.g., when a new OAM cell is generated), the 5 octet ATM cell header is represented by zero octets, except for the 3 bit PTI field. The remainder of the Cell Header fields (VPI/VCI values) is filled by the ATM entity in the FUNI network side because it is the entity that manages those values.

[Note that this specification is **not backwards compatible** with the first release of the FUNI specification, af-saa-0031.000. It is as important to note that this should not cause any interoperability issues due to the fact that there are no known implementations that are based on the first document release which did not provide sufficient specifications to enable interoperable implementations in the first place].

10.1.2 Transmission of an OAM Cell

Using the FUNI service primitives to define the encapsulation of an OAM cell in a FUNI data link frame is straightforward given the definition of the FUNI service. Figure 10.1 shows the relationships among the various layer entities involved in OAM cell flows in the normal ATM environment.



Shaded Areas of the diagram are not supported in the FUNI

Figure 10.1: Interactions Between ATM- and ATMM-entities (Fig 4/I.361)

Recall that the FUNI Data Link Service is defined by Table 1 duplicated below:

	DL-FUNI-DATA.request	DL-FUNI DATA.indication
FUNI-user-data	X	X
FUNI-CLP	X	X
FUNI-EFCI	X	X
FUNI-OAM	O	O

Duplicate of Table 5.1: FUNI Data Link Layer Service Primitives and Parameters

10.1.3 ATMM Primitives and Parameters

The FUNI-OAM parameter is set whenever the FUNI-user-data parameter contains an OAM cell. The FUNI primitives must be invoked (indirectly) by the ATM Management Plane which expects to use these primitives, depicted in Table 10.1, for ATMM to ATMM interactions.

Parameter/Primitive	ATMM-DATA.request	ATMM-DATA.indication
ATM-SDU	M	M
Submitted-CLP	M	-
Received-CLP	-	O
Congestion-Indication	-	M
PHY-CEI	M	M

Table 10.1: ATMM Service Primitives and Parameters

10.2. Mapping Between FUNI and ATMM Primitives

When used with FUNI, we define an ATMM/FUNI convergence layer to map service requests between the FUNI service and the ATMM service. That definition is given in this section.

10.2.1 PDU Format

As shown in Figure 10.2, ATMM-user data SDUs are embedded in the FUNI data link frame by constructing a 5-octet ATM cell header with the requisite PTI field and placing the 48-octet ATMM SDU after it. The 53-octets are placed into the FUNI SDU data field.

Octets			
1-2	3 to 7	8 to 55	56 to 57
or 1-2	3 to 7	8 to 55	56 to 59
or 1 to 4	5 to 9	10 to 57	58 to 61
Header	OAM Header	OAM Cell Payload	FCS

Figure 10.2: FUNI Data Link Frame Carrying OAM Cells

10.2.2 Mapping the ATMM Request into the FUNI Service Primitives

As shown in Figure 9, upon receipt of a ATMM-DATA.request signal from the ATMM-user, a DL-FUNI-DATA.request primitive is formed as described below.

1. The FUNI-user-data parameter is formed by placing five octets of zero in the OAM Header field. The PTI field (octet 6, bits 4-2 of the FUNI data link frame) are formed as follows.

- For ATMM-Data.requests originating at service points associated with F5 OAM flows, the 3 bits of PTI coding in Table 3 below shall be copied into the PTI field of the OAM Cell Header part of the FUNI frame payload.

2. The “CLP” parameter is formed using the “Submitted Loss Priority” parameter of the ATMM-DATA.request signal.
3. The “EFCI” parameter is set to “0”.
4. The PHY-CEI parameter of the ATMM-Data.request signal identifies the connection end point.

Note that each instance of the protocol stack is associated with a particular connection and so “knows” the appropriate Frame Address to use (i.e., the addressing information is not passed via the ATMM service primitives). In particular, F5 OAM flows are carried using the same VPI/VCI as the VCC.

ATMM Request Point	PTI Coding (bits 4-2)
OAM F5 segment associated cell	100
OAM F5 end-to-end associated cell	101

Table 10.2: PTI codes for F5 OAM Cell Flow

10.2.3 Mapping the FUNI Service Primitives into the ATMM Indication

As shown in Figure 9, upon receipt of a DL-FUNI-DATA.indication primitive from the FUNI service with the OAM parameter set, an ATMM-Data.indication signal is formed with the parameters noted below and sent to the ATMM user:

1. The “ATM-SDU” parameter is set to the value of the DL-FUNI-Data SDU OAM Cell Payload subfield of the “FUNI-user-data” parameter of the DL-FUNI-DATA.indication primitive.
2. The “Received CLP” parameter is set to the value in the “FUNI-CLP” parameter of the DL-FUNI-DATA.indication primitive.
3. The “Congestion Indication” parameter is set to Zero.
4. The “PHY-CEI” parameter identifies the connection end point carrying the VCC.

The delivery point for the ATMM-Data.indication is determined by the PHY-CEI and the PTI encoding in the DL-FUNI-Data OAM Cell Header subfield.

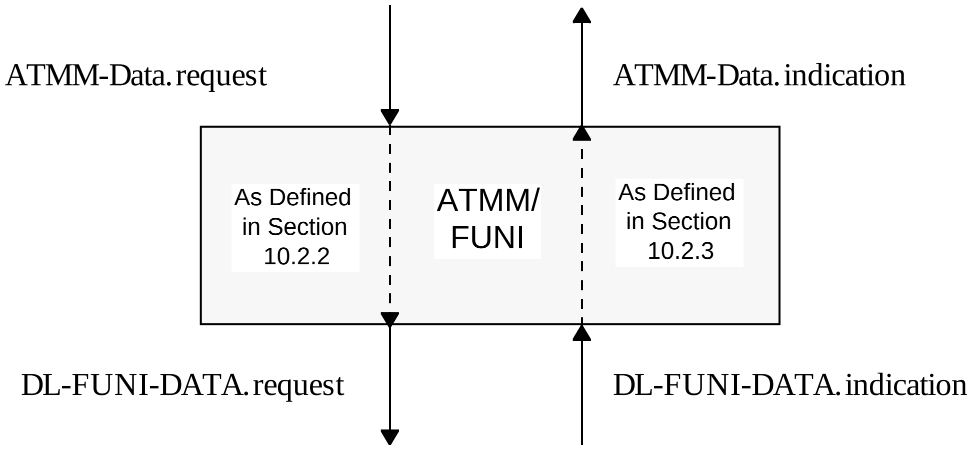


Figure 10.3: ATMM/FUNI Translation Sublayer

10.2.4 User-Side (DTE) Behavior

Transmitter: The ATMM user uses the ATMM-Data.request primitive with the ATM OAM cell payload passed as the ATM-SDU parameter, with the “Submitted CLP” parameter set to “0”, and with the “Congestion Indication” parameter set to “0”. The ATMM/FUNI convergence function passes this information to the FUNI service as defined in Section 10.2.2. The ATMM-Data.request also contains a PHY-CEI parameter, which is ignored.

Receiver: a FUNI-DATA.indication will signal the arrival of an OAM cell. The ATMM/FUNI convergence function maps the signals as defined in Section 10.2.3 into ATMM signals to deliver the DL-FUNI-Data SDU to the ATMM user. The FUNI-CLP parameter is mapped into the ATMM-Data.indication “Received CLP” parameter. The FUNI-EFCI parameter is mapped into the “Congestion Indication” parameter.

10.2.5 Network-Side (DCE) Behavior

10.2.5.1 Network to User Translation

Translation of received ATM OAM cells into FUNI Data Link PDUs uses the same mechanism described in Section 10.2.2. The FUNI-EFCI parameter is set to “0” since the "Congestion Indication" parameter of F5 OAM cells is always zero.

10.2.5.2 User to Network Translation

Translation of received FUNI Data Link PDUs into ATM OAM Cells uses the same mechanism described in Section 10.2.3.

10.3 VIRTUAL CHANNEL SERVICE

The FUNI specification requires only the support of Fault Management and Connection Verification OAM cells. The processing required when these OAM cells are received is specified in the appropriate UNI specification (e.g., ATM Forum UNI 3.1, Section 3.5.3 and in ITU-T Recommendation I.610).

(CR13) If the optional OAM frame feature of the FUNI is implemented, the network side *Shall* pass F5 OAM cells to the user-side using the procedures described in this section. The user side *Shall* process them as specified by the appropriate UNI specification and I.610 for ATM connection termination. Further processing of OAM cell flows is implementation specific (e.g., generation of network management station alarms).

(R29) The FUNI network side *Shall* be capable of performing the segment loop back function, regardless of whether the management frame feature of the FUNI is implemented or not. Note that segment loop back points must be assignable to both the FUNI network side and the FUNI user side independently.

(O7) The network side may generate VC-AIS OAM cells when it detects a connection failure or receives a failure notification from the FUNI physical layer, regardless of whether the management frame feature of the FUNI is implemented or not.

(CR14) The network side *Shall* generate VC-RDI upon receiving VC-AIS if the optional OAM frame feature is not implemented, or is implemented but not enabled.

(CR15) If the optional OAM frame feature of the FUNI is not implemented, the FUNI network-side *Shall* discard end-to-end loop back OAM cells for all VCs crossing the FUNI.

11.0 REFERENCES

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

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

[3] ANSI T1.403-1989, Carrier-to-Customer DS1 Installation Metallic Interface.

[4] ANSI T1.408-1990, ISDN Primary Rate - Customer Installation Metallic Interfaces Layer 1 Specifications.

[5] ATM Forum, DS1 Physical Layer Specifications, Version 1.0, 1994

[6] IETF RFC 1406, Definitions of the Managed Objects for the DS1 and E1 Interface Types, 1993

[7] ATM Forum, User-Network Interface Specifications, Version 3.0, 1994

[8] ATM Forum, User-Network Interface Specifications, Version 3.1, 1994

[9] ATM Forum af-sig-0061.000, User-Network Interface (UNI) Signaling Specification, Version 4.0, 1996

[10] ATM Forum af-tm-0056.000, Traffic Management Specification, Version 4.0, 1996

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

[12] IETF RFC 1695, Definition of Managed Objects for ATM Management Version 8.0 using SMPv2, August, 1994.

[13] IETF RFC 1573, Evolution of the Interfaces Group of MIB-II, January 1994.

12. ACRONYMS

CPCS	Common Part Convergence Sublayer
CPCS-CI	Common Part Convergence Sublayer - Congestion Indication
CPCS-LP	Common Part Convergence Sublayer - Loss Priority
CPCS-UU	Common Part Convergence Sublayer - User-to-User
DXI	Data eXchange Interface
EFCI	Explicit Forward Congestion Indication
FA	Frame Address
FCS	Frame Check Sequence
FUNI	Frame based User-to-Network Interface
ILMI	Integrated Local Management Interface
nrt-VBR	Non-real time Variable Bit Rate
PDU	Protocol Data Unit
PTI	Payload Type Identifier
rt-VBR	Real time Variable Bit Rate
SDU	Service Data Unit
SSCF	Service Specific Coordination Function
SSCOP	Service Specific Connection oriented Protocol
UBR	Unspecified Bit Rate
UPC	Usage Parameter Control
VBR	Variable Bit Rate

13.0 FUNI MIB

13.1 Background Information and Terminology

The FUNI defines a frame-to-cell conversion mechanism that has functions not included in the standard ATOM MIB. For example

- a FUNI Data Link frame CRC error is possible
- a Frame Address may not map into a valid VPI/VCI and vice versa
- an AAL CPCS PDU is reassembled the CRC may not check
- other AAL reassembly errors
- frames transmitted and received are not counted

The following FUNI MIB reflect these counts that are not otherwise captured.

It is beyond the scope of this MIB to define managed objects for physical layer. RFC 1406 is expected to be supported in standard compliant implementations. In addition, the interface group shall also be supported with extensions/restrictions specific to ATM FUNI.

This MIB is used in conjunction with the tables defined in RFC 1573 [13]. The following table contains specific interpretations of the application of ifTable and ifXTable objects for the FUNI interface.

Object	Meaning for FUNI Interface
ifDescr	As per the object's DESCRIPTION in RFC 1573.
ifType	The value allocated for FUNI interfaces - atmFUNI (106).
ifMTU	Set to the maximum FUNI SDU size in octets for this interface.
ifSpeed	The access rate for the FUNI interface. This could be different from the speed of the underlying physical interface, e.g., in a fractional T1 case the access rate could be 384 Kbps (the value reported in this object) whereas the speed of the underlying interface would be 1.544 Mbps (the value reported in the instance of ifSpeed for the ifEntry with type ds1).
ifPhysAddress	The primary AESA or E.164 address for this interface assigned by the FUNI interface provider. An octet string of zero length if no address is used for this interface.
ifAdminStatus	As per DESCRIPTION in RFC 1573.
ifOperStatus	As per DESCRIPTION in RFC 1573.
ifLastChange	As per DESCRIPTION in RFC 1573.
ifInOctets	The number of received octets. This includes not only the SDU, but also the FUNI header and FCS.
ifInUcastPkts	The number of non-discarded, non-errored frames received on point-to-point VCs.
ifInDiscards	The number of frames that were successfully received but were discarded because of an unrecognized VC label or insufficient buffer space. Specifically, this is the sum of the funiIfRxUnknownFaFrames and funiIfRxDiscardedFrames counters.
ifInErrors	The number of received frames that were discarded because of a reception error. Possible errors can include the FUNI frames were too long or were too short, or had FCS errors. Specifically, this is the sum of the funiIfRxAbortedFrames, funiIfRxTooShortFrames, funiIfRxTooLongFrames, and funiIfRxFcsErrFrames counters.
ifInUnknownProtos	Number of unknown or unsupported upper layer protocol frames received and discarded. This only applies to DTE interfaces, and is zero (0) for DCE interfaces.
ifOutOctets	The number of transmitted octets. This includes not only the SDU but also the FUNI header and FCS.
ifOutUcastpkts	The number of non-discarded and non-errored frames transmitted on point-to-point VCs.
ifOutDiscards	The number of frames discarded in the transmit direction. Specifically, this is equal to the funiIfTxDiscardedFrames object.

Table 13.1: ifTable and ifXTable Objects for the FUNI Interface

ifOutErrors	The number of frames discarded in the egress direction because of errors. Specifically, this is the sum of thefuniIfTxTooLongFrames, funiIfTxLenErrFrames, funiIfTxCrcErrFrames, funiIfTxPartialFrames, and funiIfTxTimeOutFrames objects.
ifName	As per DESCRIPTION in RFC 1573.
ifInMulticastPkts	The number of non-discarded and non-errored frames received on non-point-to-point VCs.
ifInBroadcastPkts	Always zero (0) as there is no broadcast frames.
ifOutMulticastPkts	The number of non-discarded and non-errored frames transmitted on non-point-to-point VCs.
ifOutBroadcastPkts	Always zero (0) as there is no broadcast frames.
ifHCInOctets	Used instead of ifInOctets when ifSpeed >=155 Mbps. See the ifInOctet item above for further details.
ifHCInUcastPkts	Used instead of ifInUcastPkts when ifSpeed is >= 155 Mbps.
ifHCInMulticastPkts	Used instead of ifInMulticastPkts when ifSpeed is >= 155 Mbps.
ifHCInBroadcastPkts	Always zero (0) as there is no broadcast frames.
ifHCOutOctets	Used instead of ifOutOctets when ifSpeed >= 155 Mbits/s.
ifHCOutUcastPkts	Used instead of ifOutUcastPkts when ifSpeed >= 155 Mbits/s.
ifHCOutMulticastPkts	Used instead of ifOutMulticastPkts when ifSpeed >= 155 Mbits/s.
ifHCOutBroadcastPkts	Always zero (0) as there are no broadcast frames.
ifLinkUpDownTrapEnable	As per DESCRIPTION in RFC 1573.
ifHighSpeed	As per DESCRIPTION in RFC 1573.
ifPromiscuousMode	Set to false(2).
ifConnectorPresent	Set to false(2).

Table 13.1: ifTable and ifXTable Objects for the FUNI Interface (continued)

The DS1/E1 ATM FUNI service MIB consists of three tables.

1. FUNI logical interface configuration table, funiIfConfTable.
2. FUNI logical interface statistics table, funiIfStatsTable.
3. FUNI virtual channel link statistics table, funiVclStatsTable.

The following defines terminology used in this MIB.

FUNI service interface - A FUNI service interface corresponds to a channel consisting of any number of time slots (1 to 24 for DS1 and 1 to 31 for E1) on a physical DS1 or E1 port. Therefore a DS1 (E1) port may support up to 24 (31) FUNI service interfaces. Other speed interfaces may also be supported.

FUNI virtual connection - A FUNI virtual connection (or simply a FUNI connection) is an SVC established via signaling or a PVC provisioned at a FUNI service interface.

FUNI logical port - A FUNI logical port is defined as a group of time slots on a DS1 or E1 port or an unchannelized interface at other speeds. A logical port corresponds to a service interface.

The terms, logical port, logical link, channel and service interface, may be used interchangeably whenever their meanings can be distinguished by the context.

13.2 FUNI MIB SPECIFICATION

```
FUNI-MIB DEFINITIONS ::= BEGIN

        IMPORTS
                MODULE-IDENTITY, OBJECT-TYPE,
                Counter32, Gauge32,
                enterprises                         FROM SNMPv2-SMI
                TEXTUAL-CONVENTION                  FROM SNMPv2-TC
                ifIndex                             FROM IF-MIB
                MODULE-COMPLIANCE, OBJECT-GROUP     FROM SNMPv2-CONF;

----------------------------------------------------------------------------
-- The FUNI MIB.

atmfFuniMIB MODULE-IDENTITY
        LAST-UPDATED    "9705080000Z"
        ORGANIZATION    "The ATM Forum"
        CONTACT-INFO
            "The ATM Forum
             2570 West El Camino Real, Suite 304
             Mountain View, CA 94040-1313 USA
             Phone: +1 415-949-6700
             Fax:   +1 415-949-6705
             info@atmforum.com"
        DESCRIPTION
            "The MIB module for managing a Version 2.0 FUNI logical interface."
        ::= { atmfFuni 1}

----------------------------------------------------------------------------
-- The object identifier subtree for the FUNI MIB.

atmForum                OBJECT IDENTIFIER ::= { enterprises 353 }
atmForumNetworkManagement       OBJECT IDENTIFIER ::= { atmForum 5 }
atmfFuni                OBJECT IDENTIFIER  ::= { atmForumNetworkManagement 6}

funiMIBObjects          OBJECT IDENTIFIER ::= { atmfFuniMIB 1 }

----------------------------------------------------------------------------
--  Textual conventions for the FUNI MIB.

    FuniValidVpi ::= TEXTUAL-CONVENTION
        STATUS        current
        DESCRIPTION
            "This object defines the valid VPI values for a FUNI logical
             interface.  Note  for Mode 1a, 1b and 3, 10 bits in the FUNI
             frame header are used to identify the FUNI frame address; 4 of
             these 10 bits map to the 4 least-significant bits in the VPI
             field of an ATM cell header, for a total of 16 VPI values in
             the limited range of 0..15.In Mode 4, 24 bits in
             the FUNI frame header are used to identify the FUNI frame
             address; 8 of these 24 bits map to the 8 bits in the VPI field
             of an ATM cell header, for a total of 256 VPI values in the
             full range of 0..255."
        SYNTAX        INTEGER (0..255)


FuniValidVci ::= TEXTUAL-CONVENTION
        STATUS        current
        DESCRIPTION
            "This object defines the valid VCI values for a FUNI logical
             interface.  Note  in Mode 1a, 1b and 3, 10 bits in the FUNI
             frame header are used to identify the FUNI frame address; 6 of
             these 10 bits map to the 6 least-significant bits in the VCI
             field of an ATM cell header, for a total of 64 VCI values in
             the limited rate of 0..63.  In Mode 4, 24 bits in
             the FUNI frame header are used to identify the FUNI frame
             address; 16 of these 24 bits map to the 16 bits in the VCI
             field of an ATM cell header, for a total of 65536 VCI values in
             the full range of 0..65535.  Also note that VCI values 0
             through 31, while considered valid, have been reserved by the
             ITU-T and the ATM Forum for various non-user applications
             (e.g., signaling, ILMI, etc.), and should be avoided for normal
             user applications."
        SYNTAX        INTEGER (0..65535)

------------------------------------------------------------------------------------
-- The FUNI MIB contains three tables which allow FUNI specific configuration
-- and access to FUNI specific statistics. In addition to the MIB groups
-- defined here access will be required to the IF-MIB and the AToM-MIB in order
-- to configure manage the physical interface and the ATM connections respecively.
--
-- FUNI V2.0 supports four different Modes of operation. Modes 1a and 1b are the same
-- as in the V1.0 specification. Modes 3 and 4 are optional and new to V2.0.
-- All modes with the exception of Mode 4 have a two byte FUNI header. Mode 4 has a
-- four byte FUNI header.
-- Modes 3 and 4 always have a 32 bit CRC and support a minimum SDU size of 9232
-- octets. These modes only support AAL5.
-- Mode 1a and 1b support a minimum SDU size of 4096 octets and either a 16 bit
-- or 32 bit CRC. Both modes support AAL5 and Mode 1b also supports AAL3/4.
--
-- The tables contained in this MIB are:-
--    FUNI Interface Configuration Table
--    FUNI Interface Statistics Table
--    FUNI Virtual Connection Statistics Table
------------------------------------------------------------------------------------

-- This is the FUNI logical interface configuration group.

funiIfConfTable OBJECT-TYPE
        SYNTAX        SEQUENCE OF FuniIfConfEntry
        MAX-ACCESS    not-accessible
        STATUS        current
        DESCRIPTION
            "A table of FUNI logical interface configuration information.
             This table provides objects not normally available in the
             IF-MIB or the ATM-MIB."
        ::= { funiMIBObjects 1 }

funiIfConfEntry OBJECT-TYPE
        SYNTAX        FuniIfConfEntry
        MAX-ACCESS    not-accessible
        STATUS        current
        DESCRIPTION
            "An entry in the FUNI logical interface configuration
             information table."
        INDEX         { ifIndex }
        ::= { funiIfConfTable 1 }


FuniIfConfEntry ::=
        SEQUENCE {
                funiIfConfMode            INTEGER,
                funiIfConfFcsBits         INTEGER,
                funiIfConfSigSupport      INTEGER,
                funiIfConfSigVpi          FuniValidVpi,
                funiIfConfSigVci          FuniValidVci,
                funiIfConfIlmiSupport     INTEGER,
                funiIfConfIlmiVpi         FuniValidVpi,
                funiIfConfIlmiVci         FuniValidVci,
                funiIfConfOamSupport      INTEGER
                }

funiIfConfMode OBJECT-TYPE
        SYNTAX        INTEGER {
                       mode1a (1),
                       mode1b (2),
                       mode3 (3),
                       mode4 (4)
                      }
        MAX-ACCESS    read-write
        STATUS        current
        DESCRIPTION
            "This object allows the mode of operation for this FUNI
             logical interface to be configured."
        DEFVAL { mode1a }
        ::= { funiIfConfEntry 1 }

funiIfConfFcsBits OBJECT-TYPE
        SYNTAX        INTEGER { fcsBits16(1), fcsBits32(2) }
        MAX-ACCESS    read-write
        STATUS        current
        DESCRIPTION
            "This object specifies whether a 16- or 32-bit FCS is used in
             each FUNI frame header over this FUNI logical interface."
        DEFVAL        { fcsBits16 }
        ::= { funiIfConfEntry 2 }

funiIfConfSigSupport OBJECT-TYPE
        SYNTAX        INTEGER { enabled(1), disabled(2) }
        MAX-ACCESS    read-write
        STATUS        current
        DESCRIPTION
            "This object specifies whether support for signaling is enabled
             or disabled on this FUNI logical interface.  A FUNI logical
             interface that cannot support signaling must only allow the
             value disabled(2).  Specifying the type of signaling used on
             a FUNI logical interface is beyond the scope of the FUNI MIB."
        DEFVAL        { disabled }
        ::= { funiIfConfEntry 3 }

funiIfConfSigVpi OBJECT-TYPE
        SYNTAX        FuniValidVpi
        MAX-ACCESS    read-write
        STATUS        current
        DESCRIPTION
            "This object specifies the VPI value used for signaling on this
             FUNI logical interface.  This object is valid only if signaling
             is supported on this FUNI logical interface."
        DEFVAL        { 0 }
        ::= { funiIfConfEntry 4 }


funiIfConfSigVci OBJECT-TYPE
        SYNTAX        FuniValidVci
        MAX-ACCESS    read-write
        STATUS        current
        DESCRIPTION
            "This object specifies the VCI value used for signaling on this
             FUNI logical interface.  This object is valid only if signaling
             is supported on this FUNI logical interface."
        DEFVAL        { 5 }
        ::= { funiIfConfEntry 5 }

funiIfConfIlmiSupport OBJECT-TYPE
        SYNTAX        INTEGER { enabled(1), disabled(2) }
        MAX-ACCESS    read-write
        STATUS        current
        DESCRIPTION
            "This object specifies whether ILMI is enabled or disabled on
             this FUNI logical interface.  A FUNI logical interface that
             cannot support ILMI must only allow the value disabled(2)."
        DEFVAL        { disabled }
        ::= { funiIfConfEntry 6 }

funiIfConfIlmiVpi OBJECT-TYPE
        SYNTAX        FuniValidVpi
        MAX-ACCESS    read-write
        STATUS        current
        DESCRIPTION
            "This object specifies the VPI value used for ILMI on this
             FUNI logical interface.  This object is valid only if ILMI
             is supported on this FUNI logical interface."
        DEFVAL        { 0 }
        ::= { funiIfConfEntry 7 }

funiIfConfIlmiVci OBJECT-TYPE
        SYNTAX        FuniValidVci
        MAX-ACCESS    read-write
        STATUS        current
        DESCRIPTION
            "This object specifies the VCI value used for ILMI on this
             FUNI logical interface.  This object is valid only if ILMI
             is supported on this FUNI logical interface."
        DEFVAL        { 16 }
        ::= { funiIfConfEntry 8 }

funiIfConfOamSupport OBJECT-TYPE
        SYNTAX        INTEGER { enabled(1), disabled(2) }
        MAX-ACCESS    read-write
        STATUS        current
        DESCRIPTION
            "This object specifies whether support for OAM cell/frame
             translation is enabled or disabled on this FUNI logical
             interface.  A FUNI logical interface that cannot support OAM
             cell/frame translation must only allow the value disabled(2)."
        DEFVAL        { disabled }
        ::= { funiIfConfEntry 9 }


----------------------------------------------------------------------------
-- This is the FUNI logical interface statistics table.

funiIfStatsTable OBJECT-TYPE
        SYNTAX        SEQUENCE OF FuniIfStatsEntry
        MAX-ACCESS    not-accessible
        STATUS        current
        DESCRIPTION
            "A table of statistical information recorded at a FUNI logical
             interface."
        ::= { funiMIBObjects 2 }

funiIfStatsEntry OBJECT-TYPE
        SYNTAX        FuniIfStatsEntry
        MAX-ACCESS    not-accessible
        STATUS        current
        DESCRIPTION
            "An entry in the FUNI logical interface statistics table."
        INDEX         { ifIndex }
        ::= { funiIfStatsTable 1 }

FuniIfStatsEntry ::=
        SEQUENCE {
                funiIfEstablishedPvccs     Gauge32,
                funiIfEstablishedSvccs     Gauge32,
                funiIfRxAbortedFrames      Counter32,
                funiIfRxTooShortFrames     Counter32,
                funiIfRxTooLongFrames      Counter32,
                funiIfRxFcsErrFrames       Counter32,
                funiIfRxUnknownFaFrames    Counter32,
                funiIfRxDiscardedFrames    Counter32,
                funiIfTxTooLongFrames      Counter32,
                funiIfTxLenErrFrames       Counter32,
                funiIfTxCrcErrFrames       Counter32,
                funiIfTxPartialFrames      Counter32,
                funiIfTxTimeOutFrames      Counter32,
                funiIfTxDiscardedFrames    Counter32
                }

funiIfEstablishedPvccs OBJECT-TYPE
        SYNTAX        Gauge32
        MAX-ACCESS    read-only
        STATUS        current
        DESCRIPTION
            "The number of permanent VCCs established on this FUNI logical
             interface."
        ::= { funiIfStatsEntry 1 }

funiIfEstablishedSvccs OBJECT-TYPE
        SYNTAX        Gauge32
        MAX-ACCESS    read-only
        STATUS        current
        DESCRIPTION
            "The number of switched VCCs established on this FUNI logical
             interface."
        ::= { funiIfStatsEntry 2 }

funiIfRxAbortedFrames OBJECT-TYPE
        SYNTAX        Counter32
        MAX-ACCESS    read-only
        STATUS        current
        DESCRIPTION
            "The number of aborted frames received on this FUNI logical
             interface.  A frame is
             considered aborted after receiving 7 or more continuous ones."
        ::= { funiIfStatsEntry 3 }

funiIfRxTooShortFrames OBJECT-TYPE
        SYNTAX        Counter32
        MAX-ACCESS    read-only
        STATUS        current
        DESCRIPTION
            "The number of frames received on this FUNI logical interface
             that were too short, and were thus discarded.  The minimum
             frame size is 5 octets."
        ::= { funiIfStatsEntry 4 }

funiIfRxTooLongFrames OBJECT-TYPE
        SYNTAX        Counter32
        MAX-ACCESS    read-only
        STATUS        current
        DESCRIPTION
            "The number of frames received on this FUNI logical interface
             that were too large, and were thus discarded."
        ::= { funiIfStatsEntry 5 }

funiIfRxFcsErrFrames OBJECT-TYPE
        SYNTAX        Counter32
        MAX-ACCESS    read-only
        STATUS        current
        DESCRIPTION
            "The number of frames received on this FUNI logical interface
             that contained FCS errors, and were thus discarded."
        ::= { funiIfStatsEntry 6 }

funiIfRxUnknownFaFrames OBJECT-TYPE
        SYNTAX        Counter32
        MAX-ACCESS    read-write
        STATUS        current
        DESCRIPTION
            "The number of frames received on this FUNI logical interface
             with an unrecognized frame address (e.g., frames that were
             received on an unknown FUNI logical interface virtual channel
             link, such as an unestablished SVC or an unprovisioned PVC."
        ::= { funiIfStatsEntry 7 }

funiIfRxDiscardedFrames OBJECT-TYPE
        SYNTAX        Counter32
        MAX-ACCESS    read-only
        STATUS        current
        DESCRIPTION
            "The number of frames received on this FUNI logical interface
             that were discarded due to any reason other than those
             described in other objects in this table (e.g., frames that
             were discarded because insufficient buffer space was available
             for storing the frame)."
        ::= { funiIfStatsEntry 8 }

funiIfTxTooLongFrames OBJECT-TYPE
        SYNTAX        Counter32
        MAX-ACCESS    read-only
        STATUS        current
        DESCRIPTION
            "The number of frames that were too large to be transmitted on
             this FUNI logical interface, and were thus discarded."
        ::= { funiIfStatsEntry 9 }


funiIfTxLenErrFrames OBJECT-TYPE
        SYNTAX        Counter32
        MAX-ACCESS    read-only
        STATUS        current
        DESCRIPTION
            "The number of reassembled frames intended to be transmitted on
             this FUNI logical interface for which the AAL5 PDU trailer
             length fields did not match the actual PDU length, and were
             thus discarded."
        ::= { funiIfStatsEntry 10 }

funiIfTxCrcErrFrames OBJECT-TYPE
        SYNTAX        Counter32
        MAX-ACCESS    read-only
        STATUS        current
        DESCRIPTION
            "The number of reassembled frames intended to be transmitted on
             this FUNI logical interface for which the AAL5 PDU, AAL3/4, or
             OAM cell had CRC errors, and were thus discarded."
        ::= { funiIfStatsEntry 11 }

funiIfTxPartialFrames OBJECT-TYPE
        SYNTAX        Counter32
        MAX-ACCESS    read-only
        STATUS        current
        DESCRIPTION
            "The number of frames which could not be completely reassembled
             for reasons other than a reassembly timeout (e.g., missing
             AAL3/4 BOM cells, etc.), and were thus discarded before being
             transmitted on this FUNI logical interface."
        ::= { funiIfStatsEntry 12 }

funiIfTxTimeOutFrames OBJECT-TYPE
        SYNTAX        Counter32
        MAX-ACCESS    read-only
        STATUS        current
        DESCRIPTION
            "The number of frames which could not be completely reassembled
             due to a reassembly timeout, and were thus discarded before
             being transmitted on this FUNI logical interface."
        ::= { funiIfStatsEntry 13 }

funiIfTxDiscardedFrames OBJECT-TYPE
        SYNTAX        Counter32
        MAX-ACCESS    read-only
        STATUS        current
        DESCRIPTION
            "The number of frames intended to be transmitted on this FUNI
             logical interface that were discarded due to any reason other
             than those described in other objects in this table (e.g.,
             frames that were discarded because insufficient buffer space
             was available for reassembling the frame)."
        ::= { funiIfStatsEntry 14 }


----------------------------------------------------------------------------
-- This is the FUNI virtual channel link statistics table.

funiVclStatsTable OBJECT-TYPE
        SYNTAX        SEQUENCE OF FuniVclStatsEntry
        MAX-ACCESS    not-accessible
        STATUS        current
        DESCRIPTION
            "A table of FUNI statistical information for individual FUNI
             virtual channel links on FUNI logical interfaces."
        ::= { funiMIBObjects 3 }

funiVclStatsEntry OBJECT-TYPE
        SYNTAX        FuniVclStatsEntry
        MAX-ACCESS    not-accessible
        STATUS        current
        DESCRIPTION
            "An entry in the FUNI virtual channel link statistics table."
        INDEX         { ifIndex,
                        funiVclFaVpi,
                        funiVclFaVci
                        }
        ::= { funiVclStatsTable 1 }
```

```
FuniVclStatsEntry ::=
        SEQUENCE {
                funiVclFaVpi              FuniValidVpi,
                funiVclFaVci              FuniValidVci,
                funiVclRxClp0Frames       Counter32,
                funiVclRxTotalFrames      Counter32,
                funiVclTxClp0Frames       Counter32,
                funiVclTxTotalFrames      Counter32,
                funiVclRxClp0Octets       Counter32,
                funiVclRxTotalOctets      Counter32,
                funiVclTxClp0Octets       Counter32,
                funiVclTxTotalOctets      Counter32,
                funiVclRxErrors           Counter32,
                funiVclTxErrors           Counter32,
                funiVclRxOamFrames        Counter32,
                funiVclTxOamFrames        Counter32
                }

funiVclFaVpi OBJECT-TYPE
        SYNTAX        FuniValidVpi
        MAX-ACCESS    not-accessible
        STATUS        current
        DESCRIPTION
            "The VPI field of this FUNI virtual channel link, as identified
             within the frame address of the FUNI frame header."
        ::= { funiVclStatsEntry 1 }

funiVclFaVci OBJECT-TYPE
        SYNTAX        FuniValidVci
        MAX-ACCESS    not-accessible
        STATUS        current
        DESCRIPTION
            "The VCI field of this FUNI virtual channel link, as identified
             within the frame address of the FUNI frame header."
        ::= { funiVclStatsEntry 2 }


funiVclRxClp0Frames OBJECT-TYPE
        SYNTAX        Counter32
        MAX-ACCESS    read-only
        STATUS        current
        DESCRIPTION
            "The number of FUNI frames with CLP=0 that were received over
             this FUNI virtual channel link."
        ::= { funiVclStatsEntry 3 }

funiVclRxTotalFrames OBJECT-TYPE
        SYNTAX        Counter32
        MAX-ACCESS    read-only
        STATUS        current
        DESCRIPTION
            "The total number of FUNI frames that were received over this
             FUNI virtual channel link."
        ::= { funiVclStatsEntry 4 }

funiVclTxClp0Frames OBJECT-TYPE
        SYNTAX        Counter32
        MAX-ACCESS    read-only
        STATUS        current
        DESCRIPTION
            "The number of FUNI frames with CLP=0 that were transmitted over
             this FUNI virtual channel link."
        ::= { funiVclStatsEntry 5 }
```

```
funiVclTxTotalFrames OBJECT-TYPE
        SYNTAX        Counter32
        MAX-ACCESS    read-only
        STATUS        current
        DESCRIPTION
            "The total number of FUNI frames that were transmitted over this
             FUNI virtual channel link."
        ::= { funiVclStatsEntry 6 }

funiVclRxClp0Octets OBJECT-TYPE
        SYNTAX        Counter32
        MAX-ACCESS    read-only
        STATUS        current
        DESCRIPTION
            "The number of octets within FUNI frames with CLP=0 that were
             received over this FUNI virtual channel link (excluding the
             opening flag, header octets, FCS octets, and closing flag).
             This is intended to count only those octets within the user
             payload of each FUNI frame received with CLP=0."
        ::= { funiVclStatsEntry 7 }

funiVclRxTotalOctets OBJECT-TYPE
        SYNTAX        Counter32
        MAX-ACCESS    read-only
        STATUS        current
        DESCRIPTION
            "The total number of octets within all FUNI frames that were
             received over this FUNI virtual channel link (excluding the
             opening flag, header octets, FCS octets, and closing flag).
             This is intended to count only those octets within the user
             payload of each received FUNI frame."
        ::= { funiVclStatsEntry 8 }


funiVclTxClp0Octets OBJECT-TYPE
        SYNTAX        Counter32
        MAX-ACCESS    read-only
        STATUS        current
        DESCRIPTION
            "The number of octets within FUNI frames with CLP=0 that were
             transmitted over this FUNI virtual channel link (excluding the
             opening flag, header octets, FCS octets, and closing flag).
             This is intended to count only those octets within the user
             payload of each of each FUNI frame transmitted with CLP=0."
        ::= { funiVclStatsEntry 9 }

funiVclTxTotalOctets OBJECT-TYPE
        SYNTAX        Counter32
        MAX-ACCESS    read-only
        STATUS        current
        DESCRIPTION
            "The total number of octets within all FUNI frames that were
             transmitted over this FUNI virtual channel link (excluding the
             opening flag, header octets, FCS octets, and closing flag).
             This is intended to count only those octets within the user
             payload of each transmitted FUNI frame."
        ::= { funiVclStatsEntry 10 }

funiVclRxErrors OBJECT-TYPE
        SYNTAX        Counter32
        MAX-ACCESS    read-only
        STATUS        current
        DESCRIPTION
            "The number of errored FUNI frames that were discarded after
```

```
             being received over this FUNI virtual channel link.  More
             detail on errors detected on individual FUNI virtual channel
             links is not available on a per-VCL basis, but is available on
             a per-interface basis via the various counters in the FUNI
             logical interface statistics table."
        ::= { funiVclStatsEntry 11 }

funiVclTxErrors OBJECT-TYPE
        SYNTAX        Counter32
        MAX-ACCESS    read-only
        STATUS        current
        DESCRIPTION
            "The number of FUNI frame reassembly processes that were aborted
             due an error before transmitting any FUNI frame over this FUNI
             virtual channel link.  More detail on errors detected on
             individual FUNI virtual channel links is not available on a
             per-VCL basis, but is available on a per-interface basis via
             the various counters in the FUNI logical interface statistics
             table."
        ::= { funiVclStatsEntry 12 }

funiVclRxOamFrames OBJECT-TYPE
        SYNTAX        Counter32
        MAX-ACCESS    read-only
        STATUS        current
        DESCRIPTION
            "The number of FUNI OAM frames (i.e., with FID1=1 and FID2=0)
             that were received over this FUNI virtual channel link."
        ::= { funiVclStatsEntry 13 }


funiVclTxOamFrames OBJECT-TYPE
        SYNTAX        Counter32
        MAX-ACCESS    read-only
        STATUS        current
        DESCRIPTION
            "The number of FUNI OAM frames (i.e., with FID1=1 and FID2=0)
             that were transmitted over this FUNI virtual channel link."
        ::= { funiVclStatsEntry 14 }

----------------------------------------------------------------------------
-- conformance information

funiMIBConformance    OBJECT IDENTIFIER ::= { atmfFuniMIB 2 }
funiMIBCompliances    OBJECT IDENTIFIER ::= { funiMIBConformance 1 }
funiMIBGroups         OBJECT IDENTIFIER ::= { funiMIBConformance 2 }

-- compliance statements

funiMIBCompliance MODULE-COMPLIANCE
        STATUS        current
        DESCRIPTION
            "The compliance statement for entities which implement the
             FUNI MIB.

             Groups of FUNI objects required for management of a FUNI
             logical interface on a FUNI network or user device are
             identified by the suffix MinGroup.

             Groups of optional FUNI objects are identified by the suffix
             OptionalGroup."

        MODULE -- this module
        MANDATORY-GROUPS {
```

```
                       funiIfConfMinGroup,
                       funiIfStatsMinGroup
                       }

        OBJECT          funiIfConfMode
        MIN-ACCESS      read-only
        DESCRIPTION
            "The ability to support modes other than 1a (and thus to specify
             whether these modes are used) is optional."

        OBJECT          funiIfConfFcsBits
        MIN-ACCESS      read-only
        DESCRIPTION
            "The ability to support a 32-bit FCS (and thus to specify
             whether a 16- or 32-bit FCS is used) is optional for Mode
             1a and 1b."

        OBJECT          funiIfConfSigSupport
        MIN-ACCESS      read-only
        DESCRIPTION
            "The ability to support signaling (and thus to specify whether
             support for it is enabled or disabled) is optional."

        OBJECT          funiIfConfSigVpi
        MIN-ACCESS      read-only
        DESCRIPTION
            "The ability to support SVCs (and thus to specify the VPI to be
             used on the signaling channel) is optional."


        OBJECT          funiIfConfSigVci
        MIN-ACCESS      read-only
        DESCRIPTION
            "The ability to support SVCs (and thus to specify the VCI to be
             used on the signaling channel) is optional."

        OBJECT          funiIfConfIlmiSupport
        MIN-ACCESS      read-only
        DESCRIPTION
            "The ability to support ILMI (and thus to specify whether it is
             enabled or disabled) is optional."

        OBJECT          funiIfConfIlmiVpi
        MIN-ACCESS      read-only
        DESCRIPTION
            "The ability to support ILMI (and thus to specify the VPI to be
             used on the ILMI channel) is optional."

        OBJECT          funiIfConfIlmiVci
        MIN-ACCESS      read-only
        DESCRIPTION
            "The ability to support ILMI (and thus to specify the VCI to be
             used on the ILMI channel) is optional."

        OBJECT          funiIfConfOamSupport
        MIN-ACCESS      read-only
        DESCRIPTION
            "The ability to support OAM cell/frame translation (and thus to
             specify whether support for it is enabled or disabled) is
             optional."

        ::= { funiMIBCompliances 1 }

-- units of conformance
```

```
funiIfConfMinGroup OBJECT-GROUP
        OBJECTS {
                  funiIfConfMode,
                  funiIfConfFcsBits,
                  funiIfConfSigSupport,
                  funiIfConfSigVpi,
                  funiIfConfSigVci,
                  funiIfConfIlmiSupport,
                  funiIfConfIlmiVpi,
                  funiIfConfIlmiVci,
                  funiIfConfOamSupport
                  }
        STATUS        current
        DESCRIPTION
            "A collection of per-FUNI logical interface configuration
             objects required for management of a minimum function FUNI
             logical interface."
        ::= { funiMIBGroups 1 }


funiIfStatsMinGroup OBJECT-GROUP
        OBJECTS {
                  funiIfEstablishedPvccs,
                  funiIfEstablishedSvccs,
                  funiIfRxAbortedFrames,
                  funiIfRxTooShortFrames,
                  funiIfRxTooLongFrames,
                  funiIfRxFcsErrFrames,
                  funiIfRxUnknownFaFrames,
                  funiIfRxDiscardedFrames,
                  funiIfTxTooLongFrames,
                  funiIfTxLenErrFrames,
                  funiIfTxCrcErrFrames,
                  funiIfTxPartialFrames,
                  funiIfTxTimeOutFrames,
                  funiIfTxDiscardedFrames
                  }
        STATUS        current
        DESCRIPTION
            "A collection of per-FUNI logical interface statistics
             objects required for management of a minimum function FUNI
             logical interface."
        ::= { funiMIBGroups 2 }

funiVclStatsOptionalGroup OBJECT-GROUP
        OBJECTS {
                  funiVclRxClp0Frames,
                  funiVclRxTotalFrames,
                  funiVclTxClp0Frames,
                  funiVclTxTotalFrames,
                  funiVclRxClp0Octets,
                  funiVclRxTotalOctets,
                  funiVclTxClp0Octets,
                  funiVclTxTotalOctets,
                  funiVclRxErrors,
                  funiVclTxErrors,
                  funiVclRxOamFrames,
                  funiVclTxOamFrames
                  }
        STATUS        current
        DESCRIPTION
            "A collection of optional per-VCL statistics objects used to
             manage individual FUNI virtual channel links."
        ::= { funiMIBGroups 3 }

END
```

End of FUNI v2.0 Specifications