Network Compatible ATM for Local Network Applications

Phase 1

Version 1.01
October 19, 1992

Document Disclaimer

This document has been prepared by engineers and scientists from four organizations: Apple Computer, Bellcore, Sun Microsystems, and Xerox. This material contains our current views on a specification of ATM for local network applications that would be compatible with ATM used in public networks. It does not necessarily represent the views of Bellcore's owners or clients, and it is not binding on Bellcore, its owners, or its clients, or on Apple Computer, Sun Microsystems, Xerox, or their respective shareholders, officers, directors, employees, or agents. The proposals are subject to change in form and numerical value after more study. The companies specifically reserve the right to add to, delete, or amend the statements contained herein at any time.

1. Introduction

This document defines a public-network compatible customer premises network based on Asynchronous Transfer Mode (ATM).

1.1 Purpose and Scope

Local Area Networks (LANs) have completed two generations of development. The first generation was typified by the CSMA/CD and Token Ring LANs standardized by the IEEE 802 committee. Second generation LANs such as FDDI have been standardized, and are now beginning deployment. Emerging multimedia applications, however, are predicted to require aggregate throughputs and real-time transport guarantees that first and second generation LANs cannot easily provide. A third generation LAN must accommodate the large volumes of traffic generated by multimedia applications.

A third generation LAN should facilitate seamless end-to-end inter-working of public and private networks. While first and second generation LAN technologies have enabled the deployment of high performance, flexible, data communications services in the local area, fundamental technology differences between LANs and the public switched networks have slowed the extension of these capabilities into the wide area. Thus, there are three primary goals for the third generation LAN:

- provide the real-time transport capabilities necessary for multimedia applications incorporating voice and video,
- provide scalable throughput that is capable of growing both per-host bandwidth (to enable applications that require large volumes of data in and out of a single host), and aggregate bandwidth (to enable installations to grow from a few to several hundred high performance hosts), and
- facilitate the inter-working between LAN and wide area network (WAN) technology.

A final explicit goal of a third generation LAN, as set forth in this document, is to take maximum advantage of existing international standards wherever possible. Thus, the majority of this document uses work done in standards as a base, and merges and adapts this work to meet the above requirements.

ATM was selected as the core technology to meet the goals outlined above. It is scalable, designed for integration of multimedia applications, and is the basis of the broadband public networks being standardized in the International Telegraph and Telephone Consultive Committee (CCITT). Below the ATM layer, the physical layer is structured to be extensible. Two different physical layers are presented in this document, a block-coded layer which builds upon technologies developed for the Fibre Channel network [14], and a SONET physical layer which is based upon technologies being deployed in the public wide area networks. Additional physical layers are anticipated to meet emerging application requirements.

Above the ATM layer, a very simple adaptation layer is supported which was designed to be efficient and easily integrated into existing higher layer protocols. Since the work on signalling in broadband standards bodies is incomplete, a permanent virtual circuit (PVC) approach is supported initially. A SNMP-based management information base (MIB) is included to allow for the creation and deletion of PVCs. The addition of signalling protocols and additional adaptation layers is expected in future versions of this document. The SNMP-based capability will, however, be retained and extended to provide other network management capabilities. Finally, the groundwork for the future addition of resource management capabilities has been laid.

1.2 Relation to International and National Standards

The network described in this document conforms, with few deviations, to the following international and national standards.

- 1990 CCITT Recommendation I.150, "B-ISDN Asynchronous Transfer Mode Functional Characteristics" [1]
- 1990 CCITT Recommendation I.361, "B-ISDN ATM Layer Specification" [2]
- 1990 CCITT Recommendation I.432 "B-ISDN User-Network Interface - Physical Layer Specification" [3]
- ANSI ATM Draft Standard, "Broadband ISDN -- ATM Layer Functionality and Specification" [7]
- ANSI Interface Draft Standard, "Broadband ISDN -- User-Network Interfaces, Rates, and Formats Specification" [8]

There are several types of requirements and options in this document, some of which deviate from these standards, and they are marked as follows:

- (CR-#) Conformant Requirement
  A requirement which conforms to the requirements of the appropriate national and international standards.
- (LR-#) Limited Requirement
  A requirement which is a sub-set of the requirements of the appropriate national and international standards.
- (DR-#) Divergent Requirement
  A requirement which diverges from the requirements of the appropriate national and international standards.
- (ER-#) Enhanced Requirement
  A requirement which is an enhancement to the requirements of the appropriate national and international standards.
- (CO-#) Conformant Option

An option which conforms to the options of the appropriate national and international standards.

- (LO-#) Limited Option
  An option which is a sub-set of the options of the appropriate national and international standards.
- (DO-#) Divergent Option
  An option which diverges from the options of the appropriate national and international standards.
- (EO-#) Enhanced Option
  An option which is an enhancement to the options of the appropriate national and international standards.

1.3 Phases of Work

This document is expected to have two phases in order to gain early insight into applications requiring the capabilities of third generation LANs, and also because the current national and international standards are still evolving. These phases are defined as:

- Phase 1: A definition of a network based upon PVCs to gain early insight into third-generation LANs. Two physical layers (block coded and SONET) are provided. A simple adaptation layer, maximally compatible with existing higher layer protocols is defined, and a SNMP capability for managing the PVCs is given.
- Phase 2: An extension to the phase 1 definition to include switched virtual circuits (SVCs), with supporting signalling protocols and adaptation layers. Also, phase 2 is expected to describe an advanced resource management scheme.

This document contains the Phase 1 definitions, and lays the foundation for the second phase.

1.4 Document Contact Information

For more information regarding this document, please contact:

Ed Geiger
Apple Computer, Inc.
20450 Stevens Creek Blvd.
MS 76-2H
Cupertino, CA 95046
(408) 974-4907 (voice)
(408) 974-0781 (fax)
edg@goofy.apple.com

Glenn Estes
Bellcore
445 South Street
Room MRE 2J-298
Morristown, NJ 07962-1910
(201) 829-4238 (voice)
(201) 829-5976 (fax)
ghe@aries.bellcore.com

W. David Sincoskie
Bellcore
445 South Street
Room MRE 2Q-286
Morristown, NJ 07962-1910
(201) 829-4426 (voice)
(201) 829-2504 (fax)
sincos@bellcore.com

Fred Sammartino
Sun Microsystems
2550 Garcia Ave.
MS MTV14-203
Mountain View, CA 94043
(415) 336-2938 (voice)
(415) 336-3156 (fax)
fred@eng.sun.com

Bryan Lyles
Xerox Palo Alto Research Center
3333 Coyote Hill Rd.
Palo Alto, CA 94304
(415) 812-4466 (voice)
(415) 812-4471 (fax)
Lyles@parc.xerox.com

This document may be obtained via anonymous FTP from the following locations. The Postscript document is named nclatm-1.01.ps and is approximately 525 kbytes in length.

A compressed version of this file, named nclatm-1.01.ps.Z is also available and is approximately 205 kbytes in length.

Host	File
thumper.bellcore.com	pub/nclatm/nclatm-1.01.ps pub/nclatm/nclatm-1.01.ps.Z
ftp.apple.com	pub/latm/nclatm-1.01.ps pub/latm/nclatm-1.01.ps.Z
parcftp.xerox.com	pub/latm/nclatm-1.01.ps pub/latm/nclatm-1.01.ps.Z

2. Local ATM Architecture

2.1 Physical Architectures

There are several components which may be used to construct a local ATM environment. They include:

- hosts,
- ATM switches,
- inter-networking devices, such as routers and gateways, and
- interfaces to the public network.

These components can be interconnected in a variety of ways. To begin, each host may be connected to one or more ATM switches. The physical interconnection between any host and the ATM switch is expected to be a point-to-point link, using an interface called I_1. The multi-access TE arrangement described in CCITT Recommendations and ANSI Standards for Broadband ISDN is not used in this document. Any host can have more than one ATM interface, connecting to different ATM switches. From the network perspective, each physical interface on this host represents a distinct destination. Therefore, for this document, each interface will be considered a distinct host. The physical interconnection between any two local ATM switches is also expected to be a point-to-point link, using an interface called I_2. Finally, the interface between the public ATM network and either a local switch or a host is also expected to be point-to-point and use an interface called I_3 which is conformant to CCITT Recommendations and ANSI Standards. There are therefore two possible host interfaces (I_1 and I_3)[1] and three possible interfaces on a switch (I_1, I_2, and I_3). The functions performed at each of these interfaces are compared in Table 1, "Comparison of Interfaces I1, I2, and I3".

Function	**Interface I_1**	**Interface I_2**	**Interface I_3**
Physical Medium	Single Mode Fiber, Multi-mode Fiber, and Twisted Pair	Single Mode Fiber, Multi-mode Fiber, and Twisted Pair	Single Mode Fiber only
Framing Format	SONET and Block-coded	SONET and Block-coded	SONET
ATM	Limited number of VPI/VCIs	Additional number of VPI/VCIs	Large number of VPI/VCIs

Table 1. Comparison of Interfaces I_1, I_2, and I_3

1. It may be possible to use an conversion box to inter-work the physical layer differences between I_1 and I_3. The higher layer differences are expected to be inter-worked at the appropriate end-points.

Function	**Interface I_1**	**Interface I_2**	**Interface I_3**
AAL	Data AAL required, AAL3-4 optional	For Phase 2 and later, a signalling AAL is required.	For Phase 2 and later, the network signalling AAL is required
Signalling (for phase 2 and later)	Local Signalling Required, Network Signalling Optional	Switch-switch signalling required	Network Signalling Required

Table 1. Comparison of Interfaces I_1, I_2, and I_3

The overall physical architecture is a mesh-star architecture, as shown in Figure 1, "Local ATM Network Example Architecture" and Figure 2, "Example Architecture of Interconnected Local ATM Networks". This mesh-connected network could span a campus environment or larger areas.

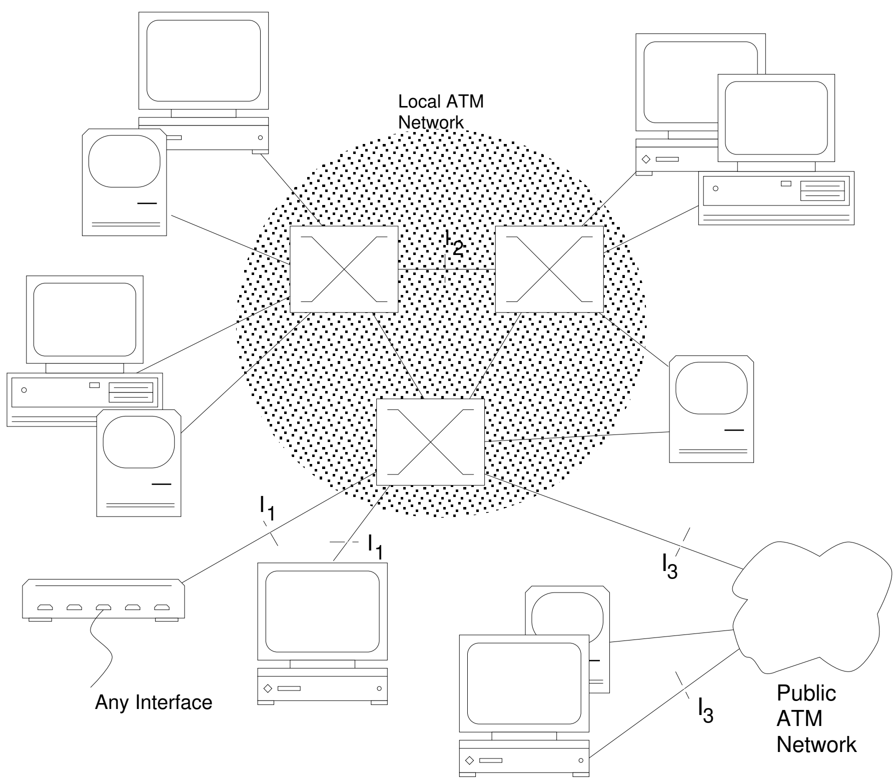


Figure 1. Local ATM Network Example Architecture

2.2 Services Provided by the Local ATM Network

On top of this mesh-star architecture, a set of network services is provided. These services include:

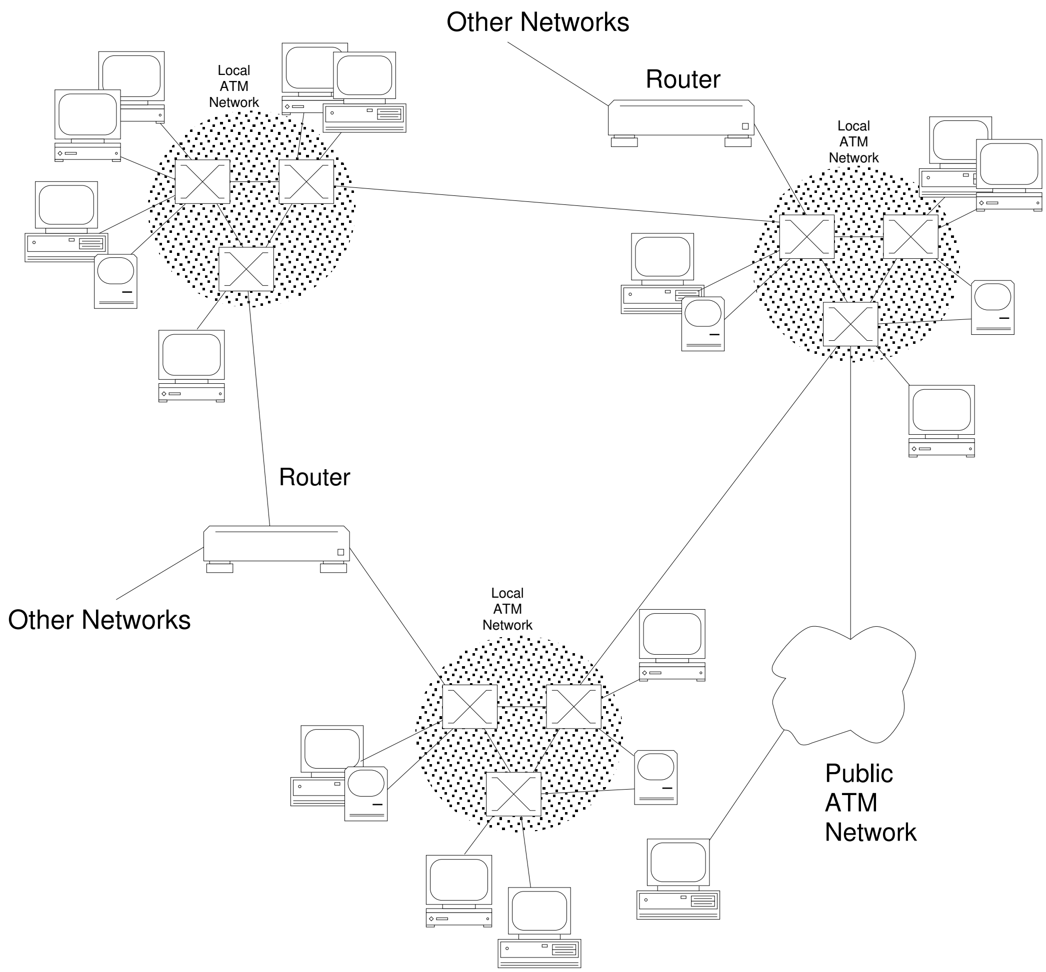


Figure 2. Example Architecture of Interconnected Local ATM Networks

- provisioned ATM connections for data transport between one transmitter and one or more receivers with a single associated adaptation layer, and
- signalled ATM connections for data transport between one transmitter and one receiver with a single associated adaptation layer.

These services rely on several ATM layer services, including:

- provisioned point-to-point ATM connections for data transport,
- provisioned point-to-multipoint ATM connections for data transport, and
- signalled point-to-point ATM connections for data transport.

The support of signalled point-to-multipoint ATM connections is **for further study**.

Central to both provisioning and signalling, a unique addressing scheme is needed for all end-points in this architecture. This addressing scheme is **for further study.**

These services are expected to support a variety of data transport protocols. The manner in which these services are used to support these data transport protocols are **for further study**.

3. Physical Layer

The Physical (PHY) layer provides for the transport of ATM cells, also known as PHY-SDUs (Service Data Units), between two ATM-entities. Additionally, this layer guarantees, within a certain probability, that only PHY-SDUs with the first five bytes free of bit errors will be delivered across the PHY Service Access Point (PHY-SAP). The PHY layer merges PHY Service Data Units (PHY-SDUs) with PHY Protocol Control Information (PHY-PCI) (i.e., the transmission overhead including the performance monitoring and alarm indication bytes, and the Header Error Control (HEC) field) to generate a continuous bit stream across the physical medium.

The physical layer provides the transmission capacity of 149.760 Mb/sec (i.e., 18.72 M bytes per second) for the ATM layer and is symmetric, i.e., it provides the same bit rate in both directions of transmission. Two transmission formats are supported: Synchronous Optical Network (SONET), and Block Coded transport. Both the SONET format and the block coded format are supported over single mode fiber, multi-mode fiber, and twisted pairs. Both LED and laser optical sources can be used for both single-mode fiber and multi-mode fiber. For multi-mode fibers, both 50 and 62.5 micron core diameters are supported. The twisted pair physical medium dependent sub-layer is **for further study**. Additional transmission media are for further study. A higher bit-rate (622.08 Mb/sec) interface will also be supported in later phases. The use of these formats is described in Table 1, "Comparison of Interfaces I1, I2, and I3".

The functions of the PHY layer have been grouped into two sub-layers: the Physical Medium Dependent (PMD) sub-layer and the Transmission Convergence (TC) sub-layer. The functions of the Physical layer are grouped into the Physical Media Dependent (PMD) sub-layer and the Transmission Convergence (TC) sub-layer

Transmission Convergence Sub-layer	HEC Generation and Verification Frame and Cell Delineation Line Coding
Physical Layer Medium Dependent Sub-layer	Bit Timing Physical Medium

Figure 3. Physical Layer Functions

as shown in Figure 3, "Physical Layer Functions". The PMD sub-layer provides bit transmission capability including symbol transfer and symbol alignment. The PMD sub-layer functions include the generation and reception of waveforms suitable for the medium, the insertion and extraction of symbol timing information, and, where appropriate, the electrical-to-optical and optical-to-electrical transformations. Both of these sub-layers communicate with the PHY Management Plane entity (PHYM-entity).

The Transmission Convergence (TC) sub-layer provides line scrambling or block coding as required, and generates and recovers transmission frames. The sending TC sub-layer entity packages the cells received from the ATM layer (PHY-SDUs) for transmission according to the payload structure of the transmission frame. It generates all the PCI of the transmission frame and the Header Error Check (HEC) on each ATM cell to be transmitted. This unit is called the TC-PDU. The receiving TC sub-layer entity extracts the cells (PHY-SDUs) from the received transmission frames (TC-PDUs) and passes them to the ATM layer only if the cell passes the HEC error detection mechanism. The TC sub-layer, also performs the maintenance and fault isolation functions for both types of TCs and cell payload scrambling/descrambling for SONET-based TCs.

3.1 Services Provided to the ATM Layer

The information exchanged between the ATM layer and the PHY layer across the PHY-SAP includes the following primitives:

Primitive	**Request**	**Indicate**	**Confirm**	**Respond**
UNITDATA	X	X		
FREQUENCY-REFERENCE		X		

Table 2. PHY Service Access Point (SAP) Primitives

In addition, the management SAP (PHYM-SAP) supports the following primitives:

Primitives	**Request**	**Indicate**	**Confirm**	**Respond**
LOSS_OF_SIGNAL		X		
LOSS-OF-FRAME		X		
LOSS-OF-CELL-DELINEATION		X		
CORRECTED-HEC		X		
UNCORRECTABLE-HEC		X		
FAR-END-RECEIVER-FAILURE		X		
FAR-END-BLOCK-ERROR		X		
ALARM-INDICATION-SIGNAL		X		

Table 3. PHYM-SAP Primitives

These primitives make use of the following parameters:

Parameter	Associated Services	Meaning	Valid Values
PHY-SDU	UNITDATA.request, UNITDATA.indicate	53 byte cell	Any 53 byte pattern
Block-Error-Count	FAR-END-BLOCK-ERROR.indicate	FEBE Information	1-24

Table 4. PHY-SAP Parameters

The primitives provide the following services:

- PHY-DATA.request
  This primitive is issued by the ATM layer to request the transfer of an ATM cell (PHY-SDU) from a local ATM-entity to the peer ATM-entity over an existing PHY-connection. In the absence of error, the entire ATM cell is transported by the PHY layer via the corresponding PHY-connection. The HEC may be modified.
- PHY-DATA.indication
  This primitive is issued to indicate the arrival of a PHY-SDU from a peer PHY-entity over an existing PHY-connection. In the absence of error, this PHY-SDU is the same PHY-SDU sent in a PHY-DATA.request primitive by the corresponding ATM peer entity. The HEC may have been modified.
- PHYM-LOSS-OF-SIGNAL.indication:
  This primitive is issued to indicate to the PHYM-entity that the receiver has lost the incoming signal and that all information can be expected to be lost because of this event.
- PHYM-LOSS-OF-FRAME.indication:
  This primitive is issued to indicate to the PHYM-entity that the receiver has lost track of the PHY frame structure and that cells and frequency reference can be expected to be lost because of this event.
- PHYM-LOSS-OF-CELL-DELINEATION.indication:
  This primitive is issued to indicate to the PHYM-entity that the receiver has lost track of the cell boundaries and that cells can be expected to be lost because of this event.
- PHYM-CORRECTED-HEC.indication
  This primitive is issued to notify a PHYM-entity that a cell was received with a detected error and that an effort was made to correct this before passing the cell to the ATM layer.
- PHYM-UNCORRECTABLE-ERROR.indication:
  This primitive is issued to notify a PHYM-entity that a cell was received with an uncorrectable error.
- PHYM-FAR-END-RECEIVER-FAILURE.indication:
  This primitive is issued to notify the PHYM-entity that the far end has declared that

its receiver has failed.

- PHYM-FAR-END-BLOCK-ERROR.indication:
  This primitive is issued to notify the PHYM-entity that one or more errors were detected over the previous frame. The number of errors are indicated in the Block-Error-Count parameter.
- PHYM-ALARM-INDICATION-SIGNAL.indication:
  This primitive is issued to notify the PHYM-entity that an upstream PHY-entity has lost its upstream signal.

The following parameters are passed within one or more of the previous primitives:

- PHY-SDU:
  This parameter indicates the 53 bytes of each ATM cell to be transferred between peer communicating ATM-entities.
- Block-Error-Count:
  The parameter indicates the approximate number of errors detected in the previous transmission frame. The exact interpretation of this number is dependent on the TC and PMD used.

3.2 PHY Layer Functions

A PHY connection can pass through many types of transmission equipment. These include originating end-systems, one or more intermediate-systems and a final terminating end-system. Intermediate-systems may or may not terminate the same TC at both interfaces. An intermediate system is shown in Figure 4, “Example Intermediate System” with

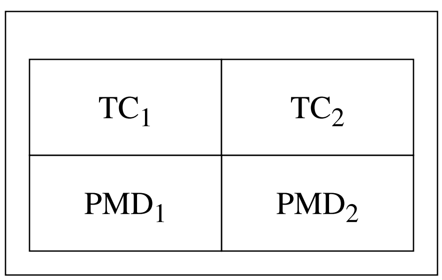


Figure 4. Example Intermediate System

two interfaces and their corresponding TC sub-layers, TC_1 and TC_2. When TC_1 and TC_2 at an intermediate system are the same, the long-term average bit-rate is expected to be the same. When TC_1 and TC_2 differ, long-time average equivalent bit-rates from input to output may drift and the intermediate system must perform bit-rate decoupling. Similarly, when TC_1 and TC_2 differ, the error characteristics are also expected to differ and therefore the intermediate system may need to terminate the HEC processing in order to ensure proper error handling. Therefore, two types of intermediate-systems will be discussed, equivalent TC intermediate systems and non-equivalent TC intermediate system.

The PHY layer is expected to perform the following functions:

- Data Transmission
  Functions related to transmission of data on the physical medium, which includes the encoding of the data into suitable electrical/optical waveforms for transmission on the medium.
- Data Reception
  Functions related to the reception of electrical/optical waveforms and their conversion into the appropriate data.
- 8 kHz Frequency Reference Transmission
  Each transmitted TC PDU shall contain a periodic pattern which conveys an 8 kHz frequency reference to a receiver. This reference may or may not be traceable to a single unique reference.
- 8 kHz Frequency Reference Reception
  This function extracts the 8 kHz frequency reference information from the data stream.
- Cell Delineation
  This function identifies the PHY-SDU (i.e., cell) boundaries within a bit stream.
- Header Error Check (HEC) Generation
  This function calculates the HEC over the PHY-SDU before transmission.
- Header Error (HEC) Processing
  Calculates the HEC over the delineated PHY-SDUs. This function detects errors in the cell header and, depending on the medium's error characteristics, may perform single bit error correction.
- Generate Performance Monitoring Information
  This function is responsible for generating information necessary for the far-end to calculate the approximate block error rate of the transmission system.
- Process Performance Monitoring Information
  This function is responsible for calculating the approximate block error rate conveying it to the far end. At the same time, this function monitors for reports of block errors from the far end and passes them to the PHYM-entity.
- Payload Rate Matching
  Facilitates the inter-working between different transport formats at the physical layer, and ensures that the physical layer provides the maximum bit rate of 149.760 Mb/s for transport of ATM cells to its ATM layer.

3.3 SONET Transmission Convergence (TC) Sub-layer

The SONET TC sub-layer, for the 155.520 Mb/s interface, is based on the SONET STS-3c structure defined in ANSI Standard T1.105 [6] and in Bellcore Technical Reference TR-

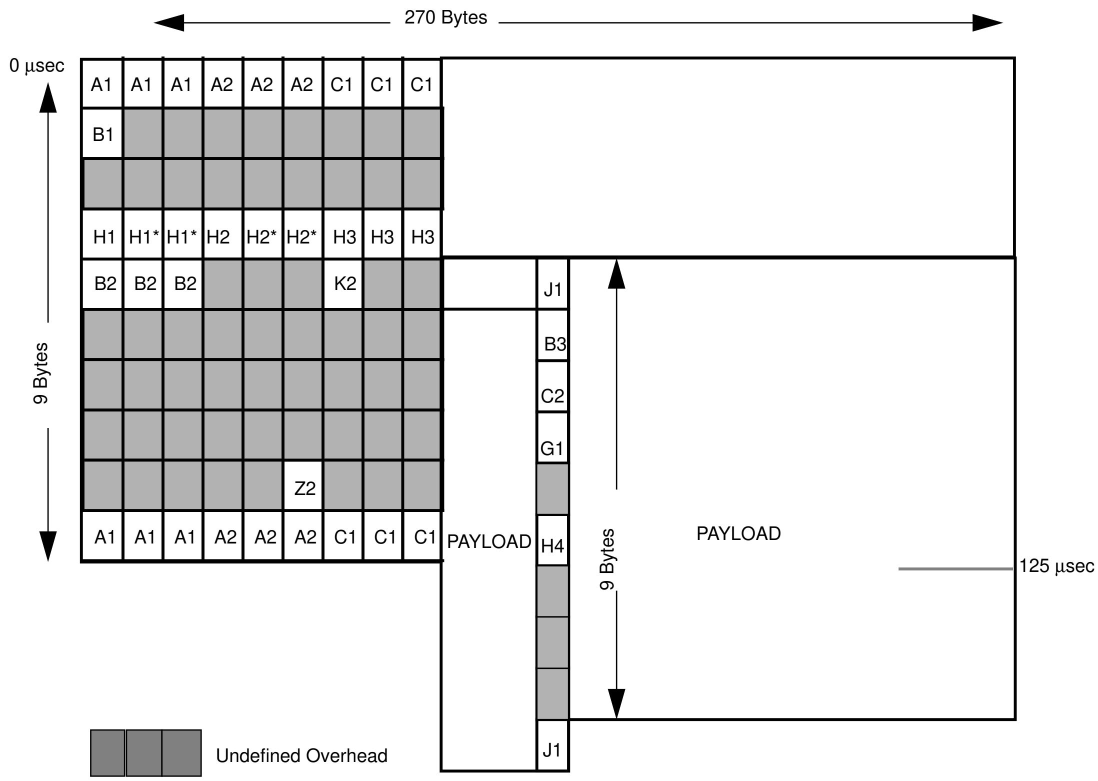


Figure 5. SONET STS-3c Frame Structure

NWT-000253 [10]. This interface provides cell transport at 149.760 Mb/sec and an information payload transport rate of approximately 135.632 Mb/sec. Figure 5, “SONET STS-3c Frame Structure” shows the structure of the SONET TC PDU (a SONET STS-3c frame) which is conformant to ANSI Standards, specifically the ANSI SONET Standard [6] and the draft standard on Broadband ISDN Interfaces and Formats [8]. Table 5, “Coding of Active SONET Overhead” shows the subset of the SONET overhead bytes which are required to be activated. The procedures for their use can be found in the

SONET Layer	Overhead Byte(s)	Function	Coding
Section Over-head	A1	Frame Alignment	11110110
	A2	Frame Alignment	00101000
	C1	STS-3c identifier	00000001 - 00000011
	B1	Section Error monitoring	BIP-8
Line Overhead	B2	Line error monitoring	BIP-24
	H1 (bits 1-4)	New Data Flag/Path AIS[a]	Normal NDF =0110, Asserted NDF = 1001
	H1 & H2 (bits 7 - 16)	Pointer value/ Path AIS[a]	0000000000 - 1100001110
	H1* H2*	Concatenation Indicator / Path AIS[a]	10010011 11111111
	H3	Pointer Action	00000000
	K2 (bits 6,7,8)	Line AIS/ Line FERF/ Remove Line FERF	111/110/ any non 110 pattern - respectively
	Z2 (3rd Z2)	Line FEBE	B2 Error Count
Path Overhead	J1	Trace	00000000
	B3	Path Error Monitoring	BIP-8
	C2	Path Signal Label	00010011
	G1 (bits 1-4)	Path FEBE	B3 Error Count
	G1 (bit 5)	Path Remote Alarm Indica-tion (RAI)	“1” to set, “0” to remove
	G1 (bits 1-4)	Path FERF	1001
	H4 (bits 3 -8)	Cell Offset Indicator	000000- 110100

Table 5. Coding of Active SONET Overhead[b]

a. Path AIS is detected by an all 1s condition in H1, H2, H1*, and H2*.

b. In this table, the MSB of each byte is numbered bit 1 and the LSB of each byte is numbered 8. The MSB is transmitted first.

SONET Standard T1.105[6]

The following is a synopsis of the SONET standard. It is a 2430 byte pattern (270 Columns by 9 Rows) which repeats every 125 µs. Transmission is row by row from left to right. The first 9 columns are the Transport Overhead (TOH) used for framing, multiplexing, and OAM functions. The remaining 2349 bytes are the Synchronous Payload Envelope (SPE) which includes 9 bytes (1 column) of Path Overhead (POH) and 2340 bytes for user payload. The SPE is defined to start at POH byte J1 and normally will begin in one frame and end in the next frame as shown. The location of the SPE (byte J1) with respect to the TOH is given by the payload pointer bytes H1 and H2.

When transporting ATM in SONET, the SPE, except for the Path overhead, is structured in blocks of 53 bytes. The H4 byte points to the first byte of the first block in each SPE and the fifth byte of each block contains the CRC generated by the HEC procedure over the first four bytes of that block.

The frame-synchronous scrambler is used as described in the ANSI SONET standards.

(CR-1) In normal operation, the timing for the transmitter at interface I_3 is locked to the timing received from the network. The tolerance under fault condition shall be 155.52 Mb/sec +/− 20 p.p.m.

(ER-2) In normal operation, the timing for the transmitter at interface I_1 and I_2 shall be 155.52 Mb/sec +/− 50 p.p.m.

(CR-3) A receiver shall acquire bit synchronization in less than 1 ms.

3.3.1 Header Error Check (HEC) Generation

This function provides protection for headers against transmission errors. Mathematically, the HEC value corresponding to a given header is defined by the following procedures:

1. The 32 bits of bytes 1, 2, 3 and 4 of the header are considered to be the coefficients of a polynomial M(x) of degree 31 (the first bit of the header corresponds to the x^{31} term and the last bit (not including the HEC) of the header corresponds to the x^0 term.)
2. M(x) is multiplied by x^8 and divided (modulo 2) by G(x). C(x) is added module 2 (exclusive OR) to the remainder of this division producing a polynomial R(x).

3. The coefficients of R(x) are considered to be the 8-bit sequence HEC.
4. The 8 bits of HEC are placed in the HEC field so that the coefficient of the x^7 term is the MSB and the coefficient of the x^0 term is the LSB.

(CR-4) The originating and intermediate TC sub-layer entities shall calculate the HEC over the first four bytes of the PHY-SDU received on each PHY-DATA.request and shall insert a Header Error Check byte (HEC) in place of the fifth header byte. The following generating polynomial G(x) and coset polynomial C(x) are used to specify the HEC value:

$$G(x) = x^8 + x^2 + x + 1$$

$$C(x) = x^6 + x^4 + x^2 + 1$$

3.3.2 Cell Framing Indication

ATM cells shall be mapped into all SPE byte positions except for the POH byte positions. Because the payload space of 2340 bytes per frame is not an integer multiple of the cell size, cells will overlap from one frame into the next and cell start locations in each frame will be offset from corresponding cell starts locations in adjacent frames. The pattern of cell starts will repeat every 53 frames.

(CR-5) To aid cell delineation at the terminating or intermediate TC-entities, the originating and intermediate TC-entities shall transmit a mini-pointer in the H4 POH position giving the start of the first cell header following the H4 byte.

3.3.3 Cell Framing Recovery

The location of the cell boundaries within a bit stream is obtained by determining the location at which the HEC coding rule is obeyed and, as an option, where the H4 byte indicates. Figure 6, “SONET TC Cell Delineation State Diagram” shows the cell delineation process.

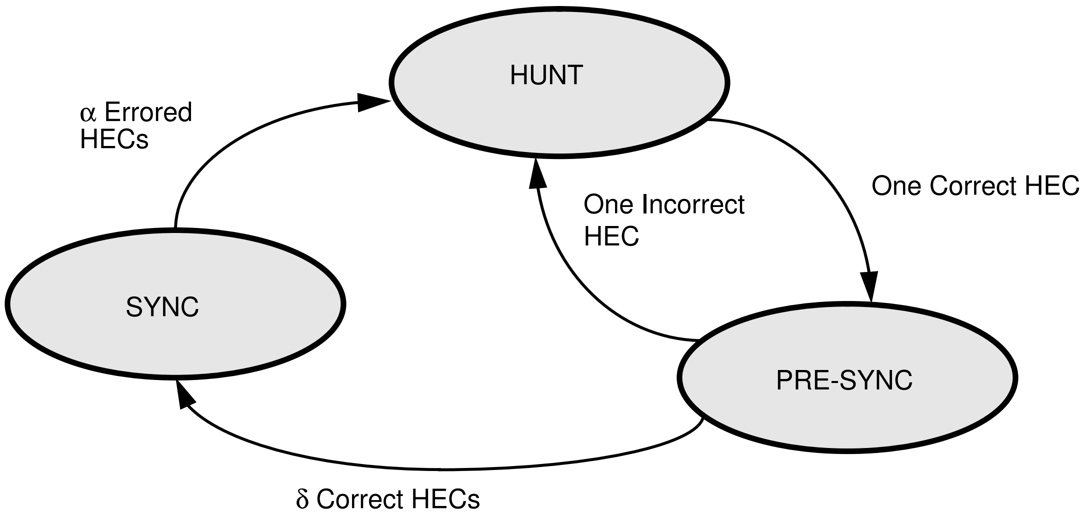


Figure 6. SONET TC Cell Delineation State Diagram

(CR-6) In the Hunt state, terminating and intermediate TC-entities shall check, bit by bit, whether the HEC coding law is respected for the assumed header field. When byte boundary can be indicated at the Physical Medium Dependent sub-layer, this process may be performed byte by byte. At detection of the first cell boundary (as indicated by a correct HEC coding) the process shall pass to the pre-sync state. While in the Hunt state, every bit received is dropped.

(CR-7) In the Pre-sync state, terminating and intermediate TC-entities shall apply the HEC rule at the next location that the cell header is expected. The process repeats until the HEC coding law has been confirmed δ consecutive times or one incorrect HEC is detected. If HEC coding law has been confirmed δ consecutive times, the process shall pass to the Sync state. If one incorrect HEC is detected the process shall return to the Hunt state. While in the Pre-sync state, every cell received is passed to the cell descrambler to allow it to begin synchronization, but these cells are not passed to the ATM layer.

(CR-8) In the Sync state, terminating and intermediate TC-entities shall apply the HEC rule at each expected cell location. Each cell received in this state is passed for further processing. When α consecutive incorrect HECs are detected, the process shall pass to the Hunt state.

Robustness against false misalignment due to bit errors depends on α. Robustness against false delineation in synchronization process depends on δ. Values of $\alpha=7$ and $\delta=6$ are suggested.

The H4 Path overhead byte gives the offset in bytes between itself and the start of the first following ATM cell. The value of H4 will vary from 0 to 52 where “0” indicates that the first byte following H4 is the first byte of an ATM header.

(CO-1) H4 may or may not be used by a receiver as an aid in the cell delineation process. H4 appears once in each frame and generally will differ in value from frame to frame.

3.3.4 Header Error Check Processing

The receiving TC entity performs the appropriate check on the HEC upon reception. For SONET systems, single-bit error correction is supported. PHY-SDUs which appear to be in error and are not correctable are discarded and not passed to the higher layer.

The receiver shall have two error handling modes as illustrated in Figure 7, “SONET TC Receiver HEC Bimodal Operation” called “Correction Mode” and the “Detection Mode.” While in “Correction Mode,” the TC sub-layer shall pass all cells with HEC syndromes of zero to the ATM layer. Cells with apparent single-bit errors will be corrected and passed to the ATM layer. Cells detected as having apparent multi-bit errors are passed through the self-synchronous scrambler and then dropped. A cell with any HEC error will cause the process to transition to the “Detection Mode.” In the “Detection Mode,” cells with non-zero syndromes are passed through the self-synchronous scrambler and then dropped. The first cell with a zero syndrome will be passed up and the process will return to the “Correction Mode.”

(CR-9) While in the “Correction mode,” the receiving PHY-entity shall check the HEC byte on each incoming PHY-SDU using the procedure described below. If the resulting syndrome is zero, the cell is passed to the higher layer. If it is not zero, an error is detected and the header is either corrected and passed to the higher layer or is discarded. If the syndrome matches one in Table 6, “Mapping of Syndromes to Single-bit Errors for SONET Systems”, then the corresponding bit is corrected, the cell is passed to the ATM layer, and the process passes to the “Detection mode.” If the syndrome is non-zero and not in Table 6, “Mapping of Syndromes to Single-bit Errors for SONET Systems”, then an apparent multi-bit error has occurred, the cell is passed to the scrambler but not passed to the higher layer. The process then passes to the “Detection mode.”

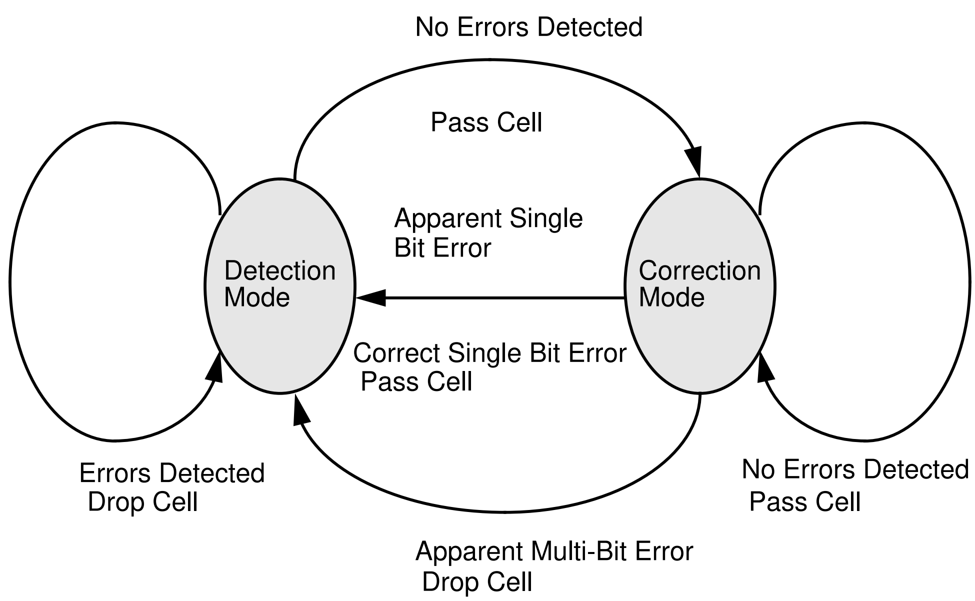


Figure 7. SONET TC Receiver HEC Bimodal Operation

(CR-10) While in the “Detection mode,” the PHY-entity shall check the HEC byte using the process described below. If the resulting syndrome is zero, the cell is passed to the ATM layer and the process passes to the “Correction mode.” If the resulting syndrome is non-zero, the cell is passed to the self-synchronous scrambler but not passed to the higher layer and the process stays in the "Detection mode."

1. The 40 bits of bytes 1, 2, 3,4 and 5 of the header are considered to be the coefficients of a polynomial M(x) of degree 39 (the first bit of the header corresponds to the x^{39} term and last bit of the HEC corresponds to the x^0 term.)
2. C(x) is subtracted modulo 2 (exclusive OR) from M(x). M(x) is then divided (modulo 2) by G(x). The remainder of this division is the syndrome S(x).

Syndrome S(x)	**Errored Bit**[a]	**Syndrome S(x)**	**Errored Bit**[a]
x^0	x^0	$x^6+x^4+x^2+x^1+x^0$	x^{20}
x^1	x^1	$x^7+x^5+x^3+x^2+x^1$	x^{21}
x^2	x^2	$x^6+x^4+x^3+x^1+x^0$	x^{22}
x^3	x^3	$x^7+x^5+x^4+x^2+x^1$	x^{23}
x^4	x^4	$x^6+x^5+x^3+x^1+x^0$	x^{24}
x^5	x^5	$x^7+x^6+x^4+x^2+x^1$	x^{25}
x^6	x^6	$x^7+x^5+x^3+x^1+x^0$	x^{26}
x^7	x^7	$x^6+x^4+x^0$	x^{27}
$x^2+x^1+x^0$	x^8	$x^7+x^5+x^1$	x^{28}
$x^3+x^2+x^1$	x^9	$x^6+x^1+x^0$	x^{29}
$x^4+x^3+x^2$	x^{10}	$x^7+x^2+x^1$	x^{30}
$x^5+x^4+x^3$	x^{11}	$x^3+x^1+x^0$	x^{31}
$x^6+x^5+x^4$	x^{12}	$x^4+x^2+x^1$	x^{32}
$x^7+x^6+x^5$	x^{13}	$x^5+x^3+x^2$	x^{33}
$x^7+x^6+x^2+x^1+x^0$	x^{14}	$x^6+x^4+x^3$	x^{34}
$x^7+x^3+x^0$	x^{15}	$x^7+x^5+x^4$	x^{35}
$x^4+x^2+x^0$	x^{16}	$x^6+x^5+x^2+x^1+x^0$	x^{36}
$x^5+x^3+x^1$	x^{17}	$x^7+x^6+x^3+x^2+x^1$	x^{37}
$x^6+x^4+x^2$	x^{18}	$x^7+x^4+x^3+x^1+x^0$	x^{38}
$x^7+x^5+x^3$	x^{19}	$x^5+x^4+x^0$	x^{39}

Table 6. Mapping of Syndromes to Single-bit Errors for SONET Systems

a. X^{39} corresponds to the first bit of the header and X^0 corresponds to the last bit of the HEC field.

3.3.5 Cell Payload Scrambling

In order to support a more robust cell delineation mechanism, the payload of every cell is scrambled using a self-synchronous scrambler. This significantly reduces the user's ability to generate false cell framing signals and the user's ability to generate long runs of consecutive ones or zeros which otherwise could force the receiver to lose bit synchronization.

(CR-11) Transmitting or intermediate TC-entity shall scramble every cell payload (last 48 bytes of each PHY-SDU) with a self-synchronous scrambler. This scrambler shall have the generator polynomial of $x^{43}+1$.

3.3.6 Cell Payload Descrambling

(CR-12) Terminating or intermediate TC-entities shall de-scramble every cell payload (last 48 bytes of each PHY-SDU) with a self-synchronous scrambler. This scrambler shall have the generator polynomial of $x^{43}+1$.

3.4 Block Coded TC Sub-layer

The Transmission Convergence (TC) sub-layer is based on the Physical layer technology developed for Fibre Channel [14]. This TC deals with physical layer aspects which are independent of the physical media characteristics. Most of the functions comprising the TC sub-layer are involved with generating and processing of some of the overhead bytes contained in the transmission format overhead and ATM cell header.

The maximum baud rate for this TC is 194.40 Mbaud/sec. This translates into a payload rate of 155.52 Mb/sec of which 149.760 Mb/sec is available for user cells. This rate has been chosen to exactly match the cell payload described in I.150 [1]. Given the ATM cell format of 5 header octets and 48 information field octets, the informational payload capacity is approximately 135.632 Mb/sec.

Refer to the paper by Widmer and Franaszek [13] for information regarding the error characteristics of this block code.

(ER-13) The transmitter shall operate at a bit-rate of 194.40 Mbaud/sec +/− 100 p.p.m.

(ER-14) A receiver shall acquire bit synchronization in less than 1 ms.

3.4.1 Line Coding

(ER-15) The 8B/10B transmission code specified in the Fibre Channel Physical layer document [14], sections 10.1 and 10.2, shall be the encoding protocol used in the Block Coded TC. Other than the K28.2, K28.5, and K28.7 Special Characters described below, use of other valid Special Characters is **for further study**.

(ER-16) The byte alignment pattern shall be the comma sequence of the 8B/10B code. The receiver shall present a properly aligned byte stream after the receipt of two K28.5 Special characters within a 5 byte window. The first byte received after the second K28.5 shall have valid byte alignment.

3.4.2 Transmission Frame Structure

The transmission frame structure used is shown in Figure 8, "Block Coded TC Frame Format" and consists of a 2430 byte frame and may vary by +/− 1 byte due to variations in the transmission rate. Within this frame, the starting position of the cells float. A sequence of 26 back-to-back cells begins 49 bytes after each K28.7 symbol, with the intervening 48 bytes comprising the overhead used to maintain the transmission system. These bytes are called the PL-OAM cells. In order to maintain the number of bytes in the frame to nominally 2430, the first PL-OAM cell after the time symbol has one less K28.5 sync symbols (three instead of four).

.

(ER-17) Figure 8, "Block Coded TC Frame Format" describes the sequence of cells in a Physical layer frame. Each sequence consists of 1 Physical layer Overhead Unit (PL-OAM unit) and 26 ATM cells. A PL-OAM unit is usually a 53 byte unit, but the

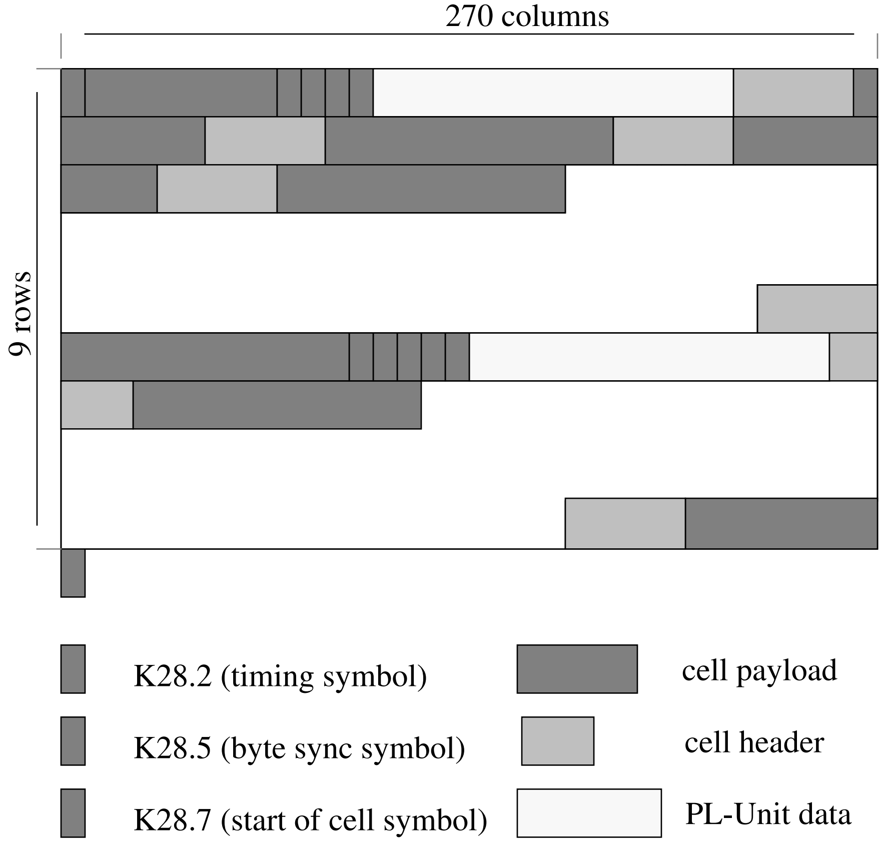


Figure 8. Block Coded TC Frame Format

first PL-OAM unit after a time symbol will have one less K28.5 symbol, making it 52 bytes long.The PL-OAM unit provides byte synchronization, frame synchronization and Physical layer OAM. This is then followed by 26 ATM layer cells. Cell rate decoupling is performed by adding unassigned cells to the data stream. The unassigned cell header is as defined in CCITT Recommendation I.361 [2]. The payload transmission rate for data cells is exactly 149.76 Mb/s (155.52 * 26/27 = 149.76).

(ER-18) The Block coded TC transmitter shall insert a time symbol (K28.2) every 125 μsec +/− 1 byte time.

(ER-19) The Block coded TC receiver shall extract the time symbols and derive the 8 kHz reference from these symbols.

The Frame Delimiter field is created from special codes to provide byte and frame synchronization. These are described in Section 3.4.3.

The Physical layer Overhead Unit is used to signal PL-OAM at the UNI. The 6th byte in the Overhead Unit contains the PL-OAM bits which are currently defined in Section 3.4.6. This byte is depicted below in Table 7, “Encoding of Byte 6 of PL-OAM Cells for Block Coded Systems”. Unused bytes within the Physical layer Overhead Unit shall be zero (0) with other values for further study.

Bit 1	**Bit 2**	**Bit 3**	**Bit 4**	**Bit 5**	**Bit 6**	**Bit 7**	**Bit 8**
FERF	EFI	AIS	0	0	0	0	0

Table 7. Encoding of Byte 6 of PL-OAM Cells for Block Coded Systems

(ER-20) The TC only passes valid, non-Physical layer cells to the ATM layer. Physical layer cells and cells with invalid HEC are not forwarded to the ATM layer.

3.4.3 Frame and Cell Delineation

Cell boundaries are synchronous with respect to frame structure. The first cell of the frame starts in the 49th byte following positive frame indication (K28.5 then K28.7 pair).

Symbol 0	**Symbol 1**	**Symbol 2**	**Symbol 3**	**Symbol 4**
K28.5	K28.5	K28.5	K28.5	K28.7

Table 8. Block Coded System Synchronization Symbols

(ER-21) Table 8, “Block Coded System Synchronization Symbols” describes the synchronization sequence used in the first five symbols of the physical layer frame. The first thre symbols are K28.5 which are used to provide byte synchronization and the fourth and fifth symbol pair , K28.5 then K28.7, provides positive indication of frame.

(ER-22) At the receiver, any cell received with any invalid data bytes in the header shall be discarded.

3.4.4 Header Error Check (HEC) Generation

This function provides protection for headers against transmission errors. Mathematically, the HEC value corresponding to a given header is defined by the following procedures:

1. The 32 bits of bytes 1, 2, 3 and 4 of the header are considered to be the coefficients of a polynomial M(x) of degree 31 (the first bit of the header corresponds to the x^{31} term and the last bit (not including the HEC) of the header corresponds to the x^0 term.)
2. M(x) is multiplied by x^8 and divided (modulo 2) by G(x). C(x) is added module 2 (exclusive OR) to the remainder of this division producing a polynomial R(x).
3. The coefficients of R(x) are considered to be the 8-bit sequence HEC.
4. The 8 bits of HEC are placed in the HEC field so that the coefficient of the x^7 term is the MSB and the coefficient of the x^0 term is the LSB.

(CR-23) The originating and intermediate TC sub-layer entities shall calculate the HEC over the first four bytes of the PHY-SDU received on each PHY-DATA.request and shall insert a Header Error Check byte (HEC) in place of the fifth header byte. The following generating polynomial G(x) and coset polynomial C(x) are used to specify the HEC value:

$$G(x) = x^8 + x^2 + x + 1$$

$$C(x) = x^6 + x^4 + x^2 + 1$$

3.4.5 Header Error Check Processing

The receiving TC entity performs the appropriate check on the HEC upon reception. For Block-coded systems, single-bit error correction is not supported. PHY-SDUs which appear to be in error are discarded and not passed to the higher layer.

(ER-24) The PHY-entity shall check the HEC byte using the process described below. If the resulting syndrome is zero, the cell is passed to the ATM layer. If the resulting syndrome is non-zero, the cell is dropped.

1. The 40 bits of bytes 1, 2, 3,4 and 5 of the header are considered to be the coefficients of a polynomial M(x) of degree 39 (the first bit of the header corresponds to the x^{39} term and last bit of the HEC corresponds to the x^0 term.)
2. C(x) is subtracted modulo 2 (exclusive OR) from M(x). M(x) is then divided (modulo 2) by G(x). The remainder of this division is the syndrome S(x).

3.4.6 Physical Layer Operation and Maintenance Procedures

The following PL-OAM functions associated with the 155.520 Mb/sec Block coded TC have been identified and are described below. These functions provide for transmission and reception of maintenance signals and low level link performance monitoring. This PL-OAM information is carried in the Physical layer Overhead Unit described in Section 3.4.2.

A maintenance signal is defined for the physical layer to indicate the detection and location of a transmission failure. This signal is:

(ER-25) Far End Receive Failure (FERF): FERF is used to alert the associated upstream termination point that a failure has been detected downstream. FERF is signaled upon the loss of frame synchronization or loss of the incoming signal. This failure is continuously indicated by a logic 1 in the Physical layer Overhead Unit until frame synchronization has been achieved.

A link transmission performance monitoring signal is defined for the physical layer to detect and report link transmission errors. This signal is used to provide a low level indication of degraded link error performance and is defined as follows:

(ER-26) Errored Frame Indicator (EFI): EFI is used to alert the associated upstream termination point that a frame has been received that contained an 8B/10B code rule violation, which can becaused by a eceived symbol not in the decoding table or a disparity error.. An EFI flag is set upon the reception of one or more encoding violations within a frame and is signaled to the upstream termination point by a logic 1 in the next available

Physical layer Overhead Unit. A set flag is cleared upon the transmission of this Physical layer Overhead Unit.

(ER-27) Alarm Indication Signal (AIS): AIS is used to alert the associated downstream termination point that a failure has been detected upstream. AIS is signaled upon the loss of frame synchronization or loss of the incoming signal. This failure is continuously indicated by a logic 1 in the Physical layer Overhead Unit until frame synchronization has been achieved.

3.5 Single-Mode Fiber PMD

The specifications of the single mode fiber PMD are summarized in Table 9, "Single-Mode PMD Parameters", and are described in the paragraphs that follow

Table 9. Single-Mode PMD Parameters

Parameter	Value
Physical Media	Two Single-Mode fiber links
Line Code	NRZ
Transmitter Wavelength	1260-1360 nm
Transmitter Maximum RMS Width	40 nm RMS
Transmitter Maximum -25 dB Width	1 nm
Transmitter Mean Launched Power	-15 to -8 dBm
Transmitter Extinction Ratio	8.2 dBm
Transmitter Eye Pattern Mask	See Figure 9, "SMF Eye Diagram for the Transmitter"
Receiver Minimum Sensitivity	-23 dBm
Receiver Maximum Overload	-8 dBm

Table 9. Single-Mode PMD Parameters

Parameter	Value
Receiver Optical Power Penalty	1 dBm

3.5.1 Fiber Specification

The physical media consists of two single mode fiber links, one for each direction of transmission.The optical parameters of the single mode fiber are specified in CCITT Recommendation G.625 [4] and Bellcore Technical Reference TR-TSY-000020, Issue 4 [11], also known as EIA class IVa fiber.

(CR-28) The physical media shall consist of two fibers meeting the EIA class IVa fiber specification.

3.5.2 Line Code

The optical line coding is binary Non-Return-to-Zero (NRZ).

3.5.3 System Budget

To ensue proper system performance, it is necessary to specify attenuation and dispersion of the optical path.

Attenuation: Attenuation shall be in the range of 0 to 7 dB. This specification is assumed to represent worst-case values including losses due to splices, connectors, optical attenuators (if used), or other passive optical devices, and any additional cable margin to cover allowances for:

- Future modifications to the cable configuration (e.g., additional splices, increased cable length, etc.),
- Fiber cable performance variations due to environmental factors, and
- Degradation of any connector, optical attenuator (if used), or other passive optical device when provided.

Dispersion: The system is assumed not to be dispersion limited.

3.5.4 Transmitter Characteristics

The feasible transmitter devices include Multi-Longitudinal Mode (MLM) lasers and Single Longitudinal Mode (SLM) lasers. It is understood that SLM devices can be substituted for MLM devices. The following parameters are specified for the transmitter.

(CR-29) The wavelength region for system operation is determined by the fiber cable characteristics; in particular, fiber attenuation. The range of the acceptable wavelength is 1260 - 1360 nm.

(CR-30) For MLM transmitter devices, the spectral width of the transmitted signal shall not exceed 40 nm RMS. The measurement method for RMS widths should take into account modes 20 dB to 30 dB down from the peak mode and is **for further study**. For SLM transmitter devices, the maximum full width of the central wavelength peak, measured 20 dB down from the maximum amplitude of the central wavelength shall not exceed 1 nm. Furthermore, for SLM devices, the minimum side-mode suppression ratio shall meet or exceed 30 db.

(CR-31) The mean launched power (MLP) in the transmit direction shall be between -15 and -8 dBm. This power is the average power of a pseudo-random data sequence coupled into the fiber by the transmitter. It is given as a range to allow for some cost optimization and to cover allowances for operation under standard operating conditions, transmitter connection degradations, measurement tolerances, and aging effects.

(CR-32) The extinction ratio shall be greater than 8.2 dB. The convention adopted for the optical level logic is:

- Emission of light for a logical “1”

- No emission for a logical “zero.”

General transmitter pulse shape characteristics including rise time, fall time, pulse overshoot, pulse undershoot, and ringing, all of which should be controlled to prevent excessive degradation of the receiver sensitivity, are specified in the form of a mask of the transmitter eye diagram.

Parameters X1, X2, and Y1 specify the normalized mask of the transmitter eye diagram. which is shown in Figure 9, “SMF Eye Diagram for the Transmitter”.

(CR-33) The transmitter, at 155.20 Mb/s rate, shall transmit within the eye diagram shown in Figure 9, “SMF Eye Diagram for the Transmitter” with parameters X1 = 0.15, X2 = 0.35, and Y1 = 0.20.

3.5.5 Receiver Characteristics

The following parameters are specified for the receiver.

(CR-34) The receiver sensitivity shall be -23 dBm or better. Receiver sensitivity is defined as the minimum value of average received power to achieve $1x10^{-10}$ bit error ratio (BER). It takes into account power penalties caused by use of a transmitter under standard operating conditions with worst-case values for extinction ratio, pulse rise and fall times, optical return loss, receiver connector degradations, and measurement tolerances. The receiver sensitivity does not include power penalties or measurement tolerances. The receiver sensitivity does not include power penalties associated with the dispersion, jitter, or reflections from the optical path; these effect are specified below in the allocation of maximum optical path penalty.

Receiver overload is the maximum acceptable value of the received average power for a $1x10^{-10}$ BER.

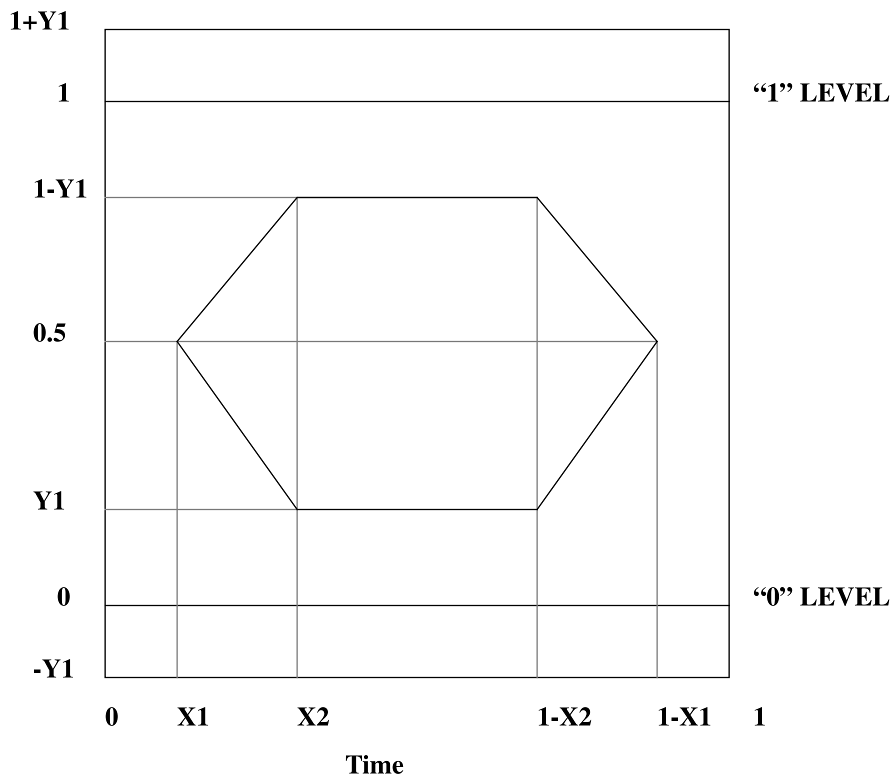


Figure 9. SMF Eye Diagram for the Transmitter

(CR-35)	The receiver overload shall be at least -8 dBm.

(CR-36)	The receiver shall be able to tolerate an optical path penalty not exceeding 1 dB to account for total degradations due to reflections, inter-symbol interference, and mode partition noise.

3.6 Multi-Mode Fiber PMD

The following PMD specification outlines the requirements for a 155.52 Mb/sec SONET or 194.4 Mbaud/sec block-coded 1300 nm multimode fiber interface. This provides for a physical interface which is a full duplex, fiber optic connection. A 62.5/125 micron, graded index, multimode fiber, with a minimum modal bandwidth of 500 MHz-km, shall be used as the communication link. Alternatively, a 50 micron core fiber may be supported as the communication link. The interface should be able to operate over a minimum of 300 meters with the 62.5/125 micron fiber, at a wavelength of 1300 nm. The minimum link length may be reduced when 50 micron fiber is incorporated.

This PMD is summarized in the following table.

Table 10. Multi-Mode Fiber PMD Parameters

Parameters	Value
Physical Media	Two Multi-Mode fiber links
Line Code	NRZ
Transmitter Wavelength	1270-1380 nm
Transmitter Spectral Width	200 nm FWHM
Transmitter Mean Launched Power	-20 to -14 dBm
Transmitter Minimum Extinction Ratio	<10%
Transmitter Pulse Mask	For Further Study
Receiver Minimum Received Power	-30 dBm
Receiver Maximum Received Power	-14 dBm
Receiver Optical Path Power Penalty	1 dB

.

3.6.1 Fiber Optic Medium Characteristics

The fiber optic medium consists of one or more sections of fiber optic cable containing one or more optical fibers as specified below along with any intermediate connectors required to connect sections together and terminated at each end in the optical connector plug. The optical fibers are interconnected to provide two continuous light paths which are connected to the port pair at each end. Each light path connects to a transmit port at one end and a receive port at the other end.

The physical media consists of two MMF links, one for each direction of transmission.

The following information regarding the MMF media takes into account the embedded base of Fiber Distributed Data Interface (FDDI) installations. Except for the wavelength operating range, the fiber parameters indicated are the same as the FDDI standard (ISO/IEC 9314-3 [12]).

This specification was developed on the basis of an attenuation value of less than or equal to 1.5 dB/km, when measured at a wavelength of 1300 nm. Higher loss fiber may be used for shorter fiber pair lengths.

(ER-37) Each optical fiber shall have a zero dispersion wavelength in the range 1270 nm to 1380 nm and a dispersion slope not exceeding 0.110 ps/nm^2-km. Each optical fiber shall have a dispersion characteristic in the range shown below.

Zero Dispersion Wavelength (Lambda)	Maximum Dispersion Slope (S_0)
1300 - 1348 nm	0.110 ps/nm^2-km
1348 - 1365 nm	[1458 - Lambda(0)] / 1000 ps/nm^2-km

Table 11. Chromatic Dispersion Requirements

(ER-38) The MMF should have a core diameter of 50 or 62.5 micron and a cladding diameter of 125 micron.The bandwidth should be at least 500MHz.km and the attenuation should be less than 1.5 dB/km at 1300 nm

(ER-39) *Attenuation Range:* The static attenuation in the optical path includes worst case values for all losses from the fiber itself, connectors, splices, attenuators or any other passive optical devices, and any additional allowance for margin, system expansion, degradation of any component, or environmental effects. The attenuation range is 0 to 9 dB.

(ER-40) *Dispersion:* The specification for fiber dispersion and transmitter spectral width accounts for the effects of dispersion to ensure correct system operation. The system is assumed not to be dispersion limited.

(ER-41) *Reflections:* The attenuation parameter includes loss due to reflections from specifications for connector forward loss. The effects on the transmitter due to reflections are assumed to be small for these bit rates and devices intended for use within the document and are, therefore, ignored.

3.6.2 Transmitter Characteristics

The values described here are for worst case and end of life; they are to be met over the full range of standard operating conditions, (i.e., voltage, temperature, and humidity), and are to include aging effects. The following parameters specify the transmitter.

(ER-42) The operating wavelength of the transmitter is specified to coincide with the operating range of the fiber. The wavelength range is from 1270 to 1380 nm.

In addition to having an embedded base of prior installations, the range centered near 1310 nm was chosen because of the low attenuation and low dispersion properties of fibers in this wavelength range and because of the trade-offs with respect to cost and technological growth.

(ER-43) The maximum spectral width is 200 nm FWHM.

(ER-44) The mean launched power (MLP) is between -14 and -20 dBm and is the average power of a pseudorandom data sequence coupled into the fiber by the transmitter.

(ER-45) The minimum extinction ratio is <10%.

(ER-46) The rise time should be between 0.6 ns to 3.0 ns.

(ER-47) The transmit optical output shall fit within the boundaries of the pulse envelope shown in Figure 10, “MMF Optical Transmit Pulse Envelope”.

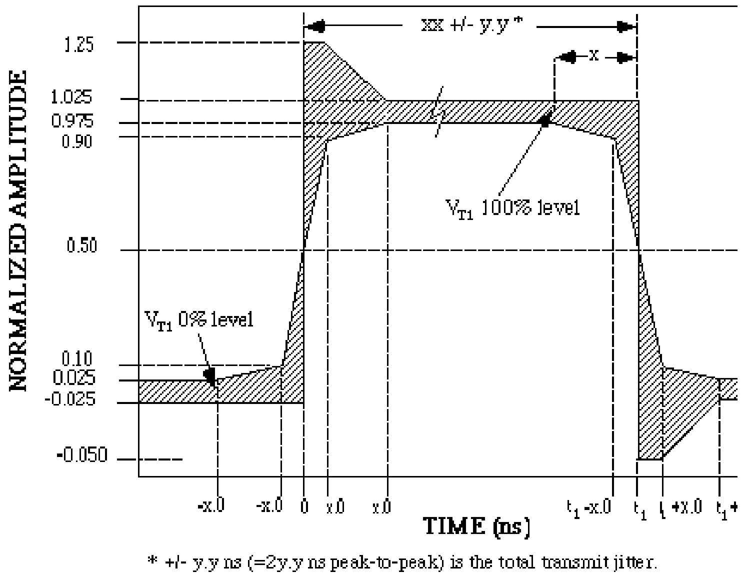


Figure 10. MMF Optical Transmit Pulse Envelope

(ER-48) The BER shall be less than 1 x 10^{-10}, when measured between physical layer interfaces attached to opposite ends of the link, for all combinations of valid optical transmit parameters,m optical receive parameters, and valid physical media.

3.6.3 Receiver Characteristics

These specifications are intended to be used for P-I-N photo-diodes. Feasible detector devices also include Avalanche Photo-Diodes (APDs), and MSM detectors. These detectors can be substituted for any applications intended for PIN detectors that do not result in any degradation in system performance.

The values prescribed are for worst case end of life; they are to be met over full range of operating conditions (i.e., voltage, temperature, and humidity), and are to include the aging effects. The following parameters are specified relating to the receiver.

(ER-49) The minimum receiver sensitivity is -29 dBm and is the minimum value of the average received power in a pseudo-random data sequence while maintaining a $1x10^{-10}$ BER.

Minimum received power takes into account receiver connector degradations and power output from the transmitter under standard operating conditions with worst case values of extinction ratio, optical pulse output dynamics (e.g., optical pulse rise and fall times), and minimum eye opening width.

(ER-50) The minimum receiver overload is -14 dBm and is the maximum value of the average received power that must be tolerated while maintaining $1x10^{-10}$BER.

(ER-51) The rise time should be between 0.6 ns to 3.0 ns.

(ER-52) The receiver is required to tolerate an optical path power penalty of 1 dB to account for degradations from fiber bandwidth limitations including dispersion, inter-symbol interference, and reflections.

3.7 Twisted Pair PMD

This work is **for further study** and expected to benefit from the work in ANSI X3T9.5 (FDDI) Twisted Pair PMDs.

3.8 Optical Connectors

3.8.1 Physical Characteristic

(ER-53) Each end of the fiber optic cable shall be terminated in connector plugs (one per fiber), as specified by the ANSI T1E1.2 Committee Draft Recommendation T1E1.2/92-020R1 for the NCLATM I_3 interface and TIA/EIA proposed recommendation 475F000 (SC Connector) for the I_1 and I_2 interfaces. (see Figure 1) The corresponding mating connector shall be used on all network elements covered by this specification to which the fiber optic cable attaches. In-line or patch panel connectors may be of other types, provided they meet the connector loss and return loss requirement below.

The use of the SC connector as an alternative to the BFOC/2.5 is **for further study**.

(ER-54) Optical connector loss shall be 0.5dB or less without application of the index matching gel.[1]. Connectors with different loss characteristisc may be used as long as any additional loss is compensated for elsewhere is the fiber loss budget.

(ER-55) The Optical connector reflectance shall be -40dB or less[2].

The connector associated with the Active Output Interface must be identified with a round white dot on both the bulkhead of the equipment and on the cable strain relief. The connector associated with the Active Input Interface must be identified with a round black dot on both the bulkhead of the equipment and on the cable strain relief. The dots must be at least 3 mm in diameter.

3.8.2 Performance Characteristics

(CR-56) Optical connector reflectance shall be -30 dB or less. Optical connector loss shall be 0.5 dB or less without the application of the index matching gel.

1. Per test method EIA/TIA 455/34, Method A (Factory Testing) or EIA/TIA-455-59 (Field Testing).
2. The number of intermediate connectors in a fiber pair may have system implications because of return loss considerations.

3.9 Electrical Connectors

This work is **for further study** and expected to benefit from the work in ANSI X3T9.5 (FDDI) Twisted Pair PMDs.

3.10 Optical Safety Requirements

(ER-57) For safety reasons, the parameters of IEC 825 Class 1 devices ("Radiation safety of laser products, equipment classification, requirements and user's guides," 1984 plus Amendment 1, 1990) should not be exceeded even under failure conditions.

If the optical power coupled into the fiber were to exceed -2dBm average, an interlock system (see CCITT Recommendation G.958 Appendix II) would be required.

4. ATM Layer

4.1 ATM Layer Services

The ATM layer provides for the transparent transfer of fixed sized ATM layer service data units (ATM-SDUs) between communicating upper layer entities (e.g., AAL-entities). Two types of connections supporting this communication are supported, point-to-point and point-to-multipoint. In the case of point-to-multipoint, a connection may have one source and multiple destinations. This transfer occurs on a pre-established ATM connection with negotiated parameters such as cell-loss ratio, cell delay, cell delay variations, throughput and traffic parameters. In phase 1, each connection is expected to be characterized by a single pair of parameters, the number of allowed cells within a period of cell times. Additional parameters may be added in later phases. Each host is expected to generate traffic which conforms to this parameter. Local ATM switches are not required to monitor/police this parameter in phase 1. The public network is expected to enforce this parameter. In future phases, additional requirements may be placed on local switches to ensure the guaranteed QOS.

The information is delivered in the order in which it is sent within each ATM connection. No retransmission of lost or corrupted information is performed by this layer. Flow control over ATM connections is **for further study**. The ATM layer also provides its users with the capability to indicate the loss priority of the data carried in each cell.

The information exchanged between the ATM layer and the upper layer (e.g., the AAL) across the ATM-SAP includes the following primitives:

Primitive	Request	Indicate	Confirm	Respond
ATM-DATA	X	X		

Table 12. ATM Service Access Point (SAP) Primitives

These primitives make use of the following parameters:

Parameter	Associated Primitives	Meaning	Valid values
ATM-SDU	ATM-DATA.request, ATM-DATA.indicate	48 byte pattern for transport	any 48 byte pattern
SDU-type	ATM-DATA.request, ATM-DATA.indicate	end-to-end cell type indicator	0 or 1

Table 13. ATM-SAP Parameters

Parameter	Associated Primitives	Meaning	Valid values
Loss-priority	ATM-DATA.request	Cell Loss-priority	high or low priority
Congestion-experienced	ATM-DATA.indicate	FECN indication	true or false

Table 13. ATM-SAP Parameters

The primitives provide the following services:

- ATM-DATA.request:
  initiates the transfer of an ATM-SDU and its associated SDU-type to its peer entity over an existing connection. The loss priority parameter and the SDU-type parameter are used to assign the proper CLP and PTI fields to the corresponding ATM-PDU generated at the ATM layer.
- ATM-DATA.indication:
  indicates the arrival of an ATM-SDU over an existing connection, along with a congestion indication and the received ATM-SDU type. In the absence of errors, the ATM-SDU is the same as the ATM-SDU sent by the corresponding remote peer upper layer entity in an ATM-DATA.request.

The following parameters are passed within one or more of the previous primitives:

- ATM-SDU:
  This parameter contains 48 bytes of ATM layer user data to be transferred by the ATM layer between peer communicating upper layer entities.
- Loss priority:
  This parameter indicates the relative importance of the information carried in the ATM-SDU.
- Congestion indication:
  This parameter indicates that the received ATM-SDU has passed through one or more network nodes experiencing congestion.
- SDU-type:
  This parameter is only used by the ATM layer user to differentiate two types of ATM-SDUs associated with an ATM connection.

4.2 Service Expected from the Physical Layer

The ATM layer expects the Physical layer to provide for the transport of ATM cells between communicating ATM-entities. The information exchanged between the ATM layer and the Physical layer across the PHY-SAP includes the following primitives:

Primitives	Request	Indicate	Confirm	Respond
PHY-UNITDATA[a]	X	X		

Table 14. PHY-SAP Services Required by ATM

a. The ATM–entity passes one cell per PHY-UNITDATA.request and accepts one cell per PHY-UNITDATA.indicate.

4.3 Functions of the ATM Layer

An ATM connection may pass through several types of ATM equipment. These include the originating ATM end-system, one or more ATM switches (called ATM intermediate-systems), and a terminating ATM end-system. When two hosts are exchanging data without the involvement of any routers, the two hosts are the ATM end-systems. If a host using an ATM interface is communicating to a host which is connected to a different type of sub-network (non-ATM), the router terminating the ATM connection and forwarding the packets is considered the terminating ATM end-system. Finally, when supporting signalling, the ATM switch itself is not considered an ATM end-system. Instead, the signalling entity is considered the ATM end-system for signalling connections.

The ATM cell consists of a 5-byte header and a 48-byte payload as shown in Figure 11, “ATM Cell Header Fields”.

The header contains the ATM layer protocol control information.

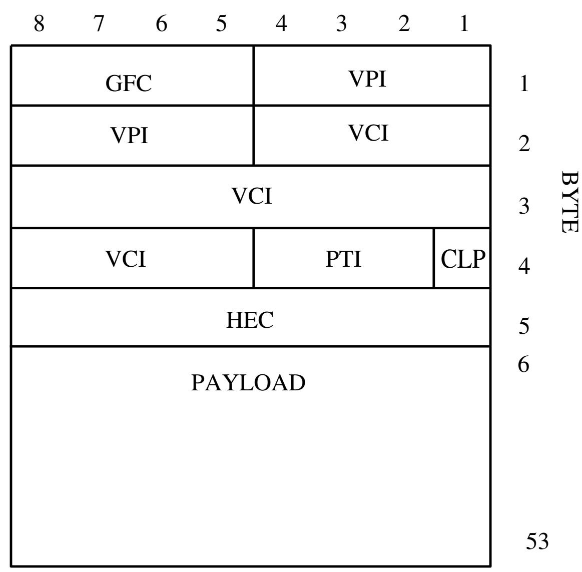


Figure 11. ATM Cell Header Fields

The actions of the ATM layer can be described in terms of several functions. These functions are described below.

4.3.1 Cell Relaying

This function forwards cells from one corresponding ATM layer entity to the corresponding ATM layer entity or entities. It is performed only at intermediate nodes in a connection. Cells can be relayed from one VP to another or from one VC to another in the same or different VP. At VP/VC switches both the VPI and VCI are translated by the relaying function. At VP switches, only the VPI is translated by the relaying function and the VCI passes through unchanged.

(CR-58) Intermediate ATM-entities shall be able to perform VC switching.

(CO-2) As an option, intermediate ATM-entities may be able to perform VP switching.

(CR-59) Intermediate ATM-entities shall be able to translate the supported receive VPI/VCI combinations into the supported transmit VPI/VCI combinations. The supported transmit VPI/VCI combinations shall conform to the following:

- the number of VPI and VCI bits may be active are **for further study**, with all inactive bits coded as zeros,
- the active bits of the VPI subfield shall be contiguous,
- the active bits of the VPI subfield shall be the least significant bits of the VPI subfield, beginning at bit 5 of byte 2,
- the active bits of the VCI subfield shall be contiguous, and
- the active bits of the VCI subfield shall be the least significant bits of the VCI subfield, beginning at bit 5 of byte 4.

(LR-60) Intermediate ATM-entities are allowed the following restrictions on the VPI/VCI combinations which they are able to receive:

- the number of VPI and VCI bits may be active are **for further study**, with all inactive bits coded as zeros,
- the active bits of the VPI subfield shall be contiguous,
- the active bits of the VPI subfield shall be the least significant bits of the VPI subfield, beginning at bit 5 of byte 2,
- the active bits of the VCI subfield shall be contiguous, and
- the active bits of the VCI subfield shall be the least significant bits of the VCI subfield, beginning at bit 5 of byte 4.

4.3.2 Cell Multiplexing

The cell multiplexing function aggregates cells from individual VC/VPs into a composite flow of cells, and is associated with the sending ATM layer entity. Individual connections are identifiable by their VPI and VCI fields.

(CR-61) Originating and intermediate ATM-entities shall be able to encode the following combinations of VPI and VCI fields:

- the number of VPI and VCI bits may be active are **for further study**, with all inactive bits coded as zeros,
- the active bits of the VPI subfield shall be contiguous,
- the active bits of the VPI subfield shall be the least significant bits of the VPI subfield, beginning at bit 5 of byte 2,
- the active bits of the VCI subfield shall be contiguous, and
- the active bits of the VCI subfield shall be the least significant bits of the VCI subfield, beginning at bit 5 of byte 4.

4.3.3 Cell Demultiplexing

The cell demultiplexing function uses the VPI, VCI and PTI to differentiate cells from various sources and deliver them to the appropriate end-point.

(LR-62) Intermediate and terminating ATM entities are allowed the following restrictions on the VPI and VCI combinations they are able to differentiate:

- the number of VPI and VCI bits may be active are **for further study**, with all inactive bits coded as zeros,
- the active bits of the VPI subfield shall be contiguous,
- the active bits of the VPI subfield shall be the least significant bits of the VPI subfield, beginning at bit 5 of byte 2,
- the active bits of the VCI subfield shall be contiguous, and
- the active bits of the VCI subfield shall be the least significant bits of the VCI subfield, beginning at bit 5 of byte 4.

4.3.4 Cell Rate Decoupling (Unassigned Cells Insertion and Deletion)

The cell rate decoupling function of the sending ATM layer entity inserts unassigned cells into the flow of assigned cells to be transmitted, transforming a non-continuous stream of assigned cells into a continuous stream of assigned and unassigned cells. The cell rate decoupling function of the receiving ATM layer entity discards the unassigned cells from the flow of cells received, transforming a continuous stream of assigned and unassigned cells into a non-continuous stream of assigned cells. Unassigned cells are identified by standardized header field values. See Section 4.4.3.

(CR-63) Originating and intermediate ATM-entities shall pass a continuous stream of assigned and unassigned cells to the PHY layer at the payload rate of the PHY layer.

(CR-64) Intermediate and terminating ATM-entities shall discard all unassigned cells as they are received from the PHY-SAP.

4.3.5 Delay Priority Processing

This function distinguishes among connections with different delay requirements, and schedules cells from these connections accordingly. The delay requirements for a given ATM connection remains constant for the duration of the connection.

(CO-3) As an option, originating and intermediate ATM-entities shall schedule cells from all connections according to their delay requirements. The delay priority of a connection is determined by its required QOS.

4.3.6 Cell Loss Priority Marking

The cell loss-priority marking function allows each ATM cell to be marked individually regarding its required cell loss-priority.

(CR-65) Originating ATM-entities shall be able to mark each ATM-SDU with either value of cell loss-priority on a per-cell basis.

4.3.7 Cell Loss-Priority Reduction

This function selectively reduces the cell loss-priority of cells which exceed their negotiated pacing rate.

(ER-66) Intermediate ATM-entities are allowed to modify the CLP field of any cell which exceed the peak rate of the connection to reduce its cell loss-priority. Specifically, CLP=0 can be altered to CLP=1 whenever the ATM-entity observes more than **i** cells within any period of **m** cell times.

4.3.8 Cell-Rate Pacing

Every connection has an associated pacing rate descriptor. Each connection is limited to transmit no more than **i** cells in **m** cell times as cells cross the PHY-SAP. The parameters **i** and **m** are set on a connection basis and may be altered during the lifetime of a connection.

(ER-67) Originating ATM-entities shall not transmit more than **i** cells within any period of **m** cell times unless the connection has been established with the ability to exceed the pacing rate. This transmission restriction is required on a per-VPI/VCI basis and over the aggregate of high CLP and low CLP traffic.

Additional transmission rate descriptors are **for further study.**

(CO-4) As an option, intermediate ATM-entities shall not transmit more than **i** cells within any period of **m** cell times unless the connection has been established with the ability to exceed the pacing rate. This transmission restriction is required on a per-VPI/VCI basis and over the aggregate of high CLP and low CLP traffic.

4.3.9 Cell-Rate Transmission Exceeding Pacing Rate

Connections can be established with the ability to exceed the pacing rate. This ability must be confirmed (via signalling or provisioning) before any source can exceed the pacing rate.

(CO-5) As an option, originating ATM-entities shall be able to transmit in excess of their pacing rate if the connection has been established with the ability to support this. The allowed shape of traffic in excess of the pacing rate is **for further study.**

4.3.10 Peak-Rate Enforcement

This function monitors each connection to identify those cells that are not in compliance with the enforcement traffic descriptor (ETD). This descriptor is derived from thepacing rate descriptor agreed upon at connection establishment. The ETD may need to account for cell delay variation that may be introduced by access multiplexing.

(CO-6) As an option, when cells in excess of the **i** cells in **m** cell times are detected, intermediate ATM-entity may or may not take actions such as:

- performing cell loss-priority reduction of high priority traffic and discarding cells in excess of the agreed pacing rate, or
- discarding non-complying cells.

4.3.11 Explicit Forward Congestion Marking

This function sets an explicit forward congestion notification (EFCN) indicator in the cell header so that this indicator may be examined at the destination. It is set by intermediate nodes in congested state. When an intermediate node is not in a congested state, it will not modify the EFCN indicator.

(CR-68) Intermediate ATM-entities, upon experiencing congestion[1], shall change PTI=000 and PTI=001 to PTI=010 and PTI=011 respectively.

4.3.12 Explicit Forward Congestion Indication To Higher Layer

---

(CR-69) Terminating ATM-entities shall indicate to the higher layer whether congestion was experienced or not for every cell in a connection.

---

1. In the context of EFCN, the term congestion is a situation where the network anticipates a deterioration of the current QOS below the connection's QOS.

4.3.13 Cell Payload Type Marking

(CR-70) Originating and intermediate ATM-entities performing this function shall mark every cell with the correct PTI as described in Table 15, “ATM Layer Payload Type Indicator Encoding”. Originating and intermediate ATM-entities shall be capable of encoding all PTI encodings. The transmission of cells marked with PTI=100, 101, 110, and 111 is **for further study.**

4.3.14 Cell Payload Type Differentiation

This function discriminates between cells of various types.

(CR-71) Intermediate and terminating ATM-entities shall be able to differentiate between all PTI encodings as described in Table 15, “ATM Layer Payload Type Indicator Encoding”. The processing of cells marked with PTI=100, 101, 110, and 111 is **for further study.**

4.3.15 Generic Flow Control “Uncontrolled Procedures”

Two modes of operation have been defined for operation of the GFC field. These are “uncontrolled access” and “controlled access.” The “uncontrolled access” mode of operation is used in ATM LAN environments. This mode has no impact on the traffic which a host generates.

(CR-72) Each originating ATM-entity shall transmit the GFC field set to all zeros (0000).

In order to avoid unwanted interactions between this mode and the “controlled access” mode where hosts are expected to modify their transmissions according to the activity of the GFC field, all terminating ATM-entities should monitor the GFC field to ensure the attached equipment is operating in “uncontrolled mode.”

(CR-73) A count of the number of non-zero GFC fields shall be measured for non-overlapping intervals of 30,000 +/− 10,000 cell times. If ten (10) or more non-zero values are received within this interval, an error shall be indicated to Layer Management.

4.4 Cell Structure and Encoding

In this document, a single cell format, referred to in ANSI Standards as the UNI cell format, is used.

4.4.1 Encoding Principles

A field is encoded with its MSB in the highest number bit of the lowest number byte that the field spans. The remaining bits of the field, in progressively decreasing significance, are placed in decreasing bit positions within the same byte, then in increasing bytes. These encoding conventions are illustrated in Figure 12, “ATM Field Encoding Convention”.

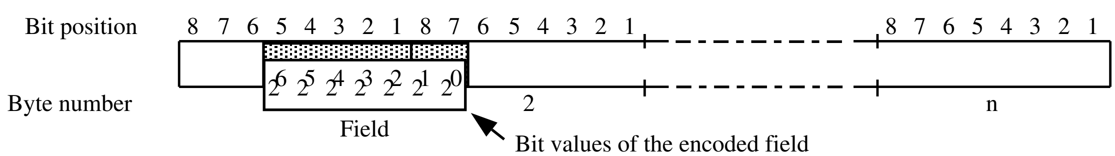


Figure 12. ATM Field Encoding Convention

4.4.2 Cell Structure

The ATM cell contains the following fields:

- Generic Flow Control (GFC). A 4-bit field used by the generic flow control mechanism. The encoding of this field is all zeros.
- VPI/VCI field. A 24-bit field containing 2 subfields:

  an 8-bit VPI, and

  a 16-bit VCI.
- Payload Type (PTI) field. A 3-bit field used to indicate whether a cell contains upper layer information, end-to-end Layer Management information, or segment Layer Management information in the payload. Table 15, “ATM Layer Payload Type

Indicator Encoding" describes the PTI field encoding. Cell Loss-priority (CLP)

PTI Coding (MSB first)	Interpretation
0 0 0	User data cell, congestion not experienced, SDU-type=0
0 0 1	User data cell, congestion not experienced, SDU-type=1
0 1 0	User data cell, congestion experienced, SDU-type=0
0 1 1	User data cell, congestion experienced, SDU-type=1
1 0 0	Segment OAM F5 flow related cell
1 0 1	End-to-end OAM F5 flow related cell
1 1 0	Resource management cell
1 1 1	Reserved for future functions

Table 15. ATM Layer Payload Type Indicator Encoding

field. A 1-bit field used by the Cell Loss-priority Marking function. If the CLP is set (CLP value is 1), the cell is subject to discard, depending on network conditions. If the CLP is not set (CLP value is 0), the cell has higher priority.

- Header Error Control (HEC) field. An 8-bit field used by the Header Error Check generation and processing functions of the PHY layer.

4.4.3 Pre-assigned Header Field Values

Pre-assigned header field values are given in Table 16, "ATM Layer Pre-assigned Header Values", excluding the HEC field. The VCI value of zero is not available for user Virtual Channel identification. ATM cells containing invalid patterns are discarded by the ATM layer.

Use	Value[a b c]			
	byte 1	**byte 2**	**byte 3**	**byte 4**
Unassigned cell indication	00000000	00000000	00000000	0000AAA0
Meta-signalling (default) (phase 2)	00000000	00000000	00000000	0001AAAA
Meta-signalling (phase 2)	0000YYYY	YYYY0000	00000000	0001AAAA
General Broadcast signalling (default) (phase 2)	00000000	00000000	00000000	0010AAAA
General Broadcast signalling (phase 2) [d]	0000YYYY	YYYY0000	00000000	0010AAAA
Invalid Pattern	XXXX0000	00000000	00000000	0000XXX1

Table 16. ATM Layer Pre-assigned Header Values

Use	Value[a b c]			
	byte 1	**byte 2**	**byte 3**	**byte 4**
End-to-end or Segment OAM F4 Flow cell	0000AAAA	AAAA0000	00000000	0011AAAA
ES-to-ES OAM F4 Flow cell	0000AAAA	AAAA0000	00000000	0100AAAA

Table 16. ATM Layer Pre-assigned Header Values

a. A - indicates that the bit is available for use by the appropriate ATM layer function.

b. X - indicates “don’t care” bits.

c. Y - any VPI value other than 00000000.

d. For ATM LAN configurations (which have only a single workstation per switch port) the General Broadcast Signalling Channel may be used to support host-switch signalling.

5. Data Transfer AAL

The purpose of the ATM Adaptation Layer (AAL) is to provide those capabilities necessary to meet the user layer data transfer needs while using the service of the ATM layer. This protocol provides the transport of variable length frames (up to 65535 bytes in length) with error detection[1]. The frame is padded to align the resulting protocol data unit to fill an integral number of ATM cells. A length field is used to extract the frame and detect additional errors not detected by the CRC-32. This adaptation protocol is designed for efficient implementation in hardware.

This AAL can be viewed as providing a capability similar to that provided by the AAL3/4 common part described in the ANSI draft standard [17] and CCITT Recommendation I.363 [18] and CCITT Draft Recommendation . It builds on the work presented to Committee T1 in contribution T1S1.5/91-449 [20].

5.1 Service Provided to the User Layer

The AAL provides the capability to transfer variable length (up to 65535 bytes) byte aligned AAL-SDUs from one AAL user to one or more AAL users. During this process, AAL-SDUs may be lost or corrupted. Lost or corrupted AAL-SDUs will not be recovered by the AAL. As an option, corrupted AAL-SDUs may be delivered to the AAL user with an indication of the error. The implementation of errored delivery is an option. If the option is supported by an implementation, the use of the errored delivery service is an optional service which should be invoked through the management plane on a per-connection basis.

The information exchanged between the AAL layer and the upper layer across the AAL-SAP includes the following primitives:

Primitive	**Request**	**Indicate**	**Confirm**	**Response**
AAL-UNITDATA	X	X		

Table 17. Data AAL Service Access Point (SAP) Primitives

In addition, the management SAP (Data MAAL-SAP) supports the following primitives:

Primitives	**Request**	**Indicate**	**Confirm**	**Respond**
MAAL-CREATE	X			
MAAL-REMOVE	X			

Table 18. Data MAAL-SAP Primitives

1. For lengths greater than 11454 bytes, the Hamming distance of the CRC is reduced from 4 to 3.

Primitives	Request	Indicate	Confirm	Respond
MAAL-ERROR		X		
MAAL-SET-TXR	X			
MAAL-SET-NOW-TXR	X			
MAAL-SET-RCVR	X			
MAAL-RESET-TXR	X		X	
MAAL-RESET-RCVR	X		X	

Table 18. Data MAAL-SAP Primitives

One or more of these primitives makes use of the following parameters:

Parameter	Associated Primitives	Meaning	Valid Values
Data	AAL-UNITDATA.request, AAL-UNITDATA.indicate	User frame for transmission	Any pattern of length from 0 to 65535 bytes
Loss_priority	AAL-UNITDATA.request	Treatment of packet with respect to ATM loss characteristics.	High, normal, and low
Forward_congestion_indication	AAL-UNITDATA.indicate	Indication that the frame experienced congestion	**For further study**
CRC_violation	AAL-UNITDATA.indicate, MAAL-ERROR.indicate	CRC error indication	True or false
CRC_received	AAL-UNITDATA.indicate	Received CRC from transmitter	Any 32 bit pattern
CRC_result	AAL-UNITDATA.indicate	Result of the crc calculation	Any 32 bit pattern
Length_violation	AAL-UNITDATA.indicate, MAAL-ERROR.indicate	Length error indication	True or false
Length_field	AAL-UNITDATA.indicate	Received length field	0-65535
Invalid_format	AAL-UNITDATA.indicate, MAAL-ERROR.indicate	Format error indication	True or false
Reassembly_time_out	AAL-UNITDATA.indicate, MAAL-ERROR.indicate	Indication of a reassembly time-out.	True or false
Oversized_received_sdu	MAAL-ERROR.indicate	Indication of an oversized received frame	True or false
Oversized_submitted_sdu	AAL-UNITDATA.indicate, MAAL-ERROR.indicate	Indication of an oversized submitted SDU	True or false
AAL_CEI	MAAL-CREATE.request, MAAL-REMOVE.request	AAL Connection Endpoint Identifier	Implementation dependent

Table 19. Data AAL-SAP and MAAL-SAP Parameters

Parameter	Associated Primitives	Meaning	Valid Values
ATM_CEI	MAAL-CREATE.request, MAAL-REMOVE.request	ATM Connection Endpoint Identifiers	Implementation dependent
Mmax_sdu_send_length	MAAL-CREATE.request, MAAL-SET-TXR.request, MAAL-SET-NOW-TXR.request	Maximum allowed length of SDU to transmit	1-65535
Mmax_sdu_deliver_length	MAAL-CREATE.request, MAAL-SET-RCVR.request	Maximum allowed length of SDU to receive	1-65535
MT1	MAAL-CREATE.request, MAAL-SET-RCVR.request	Maximum reassembly time	**For further study**
Mdeliver_errored	MAAL-CREATE.request, MAAL-SET-RCVR.request	Enable or disable errored delivery option	True or false

Table 19. Data AAL-SAP and MAAL-SAP Parameters

The following primitives provide the following services:

- AAL-UNITDATA.request
  This primitive is received from the AAL user to request the transfer of an AAL-SDU over the associated AAL connection.
- AAL-UNITDATA.indicate
  This primitive is issued to the AAL user to indicate the arrival of an AAL-SDU from the associated connection.
- MAAL-CREATE.request
  This primitive is used to create an AAL connection. Upon reception of this primitive the transmitter and receiver state machines are created and associated with the indicated AAL_CEI (AAL Connection Endpoint Identifier) and ATM_CEI (ATM Connection Endpoint Identifier).
- MAAL-REMOVE.request
  This primitive is used to remove an AAL connection. Upon reception of this primitive the transmitter and receiver state machines for the specified connection are removed.
- MAAL-ERROR.indicate
  This primitive is used to indicate to layer management errors detected by the AAL-entity.
- MAAL-SET-TXR.request
  This primitive is used to set the connection parameters (see Section 5.2) associated with the AAL transmitter for all **subsequent** AAL-UNITDATA.request primitives. For example, if two AAL-UNITDATA.request primitives are waiting to be serviced, the updating of the parameters will not occur until after these two requests are serviced.

- MAAL-SET-NOW-TXR.request
  This primitive is used to set the transmission parameters associated with the AAL connection for the next AAL-UNITDATA.request primitive to be serviced.
- MAAL-SET-RCVR.request
  This primitive is used to set the receiver parameters associated with the AAL connection.
- MAAL-RESET-TXR.request/confirm
  This primitive is used to reset the transmitter state machine. This results in the discarding of any AAL-UNITDATA.requests and the termination of any AAL-PDUs being processed. The confirmation primitive is used to indicate that the reset has been complete.
- MAAL-RESET-RCVR.request/confirm
  This primitive is used to reset the receiver state machine. This results in the termination of any AAL-PDU being processed. The confirmation primitive is used to indicate that the reset has been complete.

The following parameters are passed within one or more of the previous two tables of primitives:

- Data
  This parameter specifies the AAL-SDU transferred between the AAL user and the AAL. This parameter is byte aligned and can range from 0 to 65535 bytes in length.
- Loss-priority
  This parameter indicates the loss priority assigned to the AAL-SDU. Three levels of priority are identified: high, normal, and low. AAL-SDUs submitted with a high loss priority value will be sent in cells that are all marked with a high cell loss priority. AAL-SDUs submitted with a normal loss priority value will be sent in cells that are all marked with a low cell loss priority except the last cell, which will be marked with a high cell loss priority. AAL-SDUs submitted with a low loss priority value will be sent in cells that are all marked with a low cell loss priority.
- Forward_congestion_indication
  This parameter indicates the detection of congestion experienced by the received AAL-SDU. The acceptable values of this parameter and its computation are **for further study**.The computation of this parameter value will use the Congestion_-experienced parameter received in the ATM-DATA.indication.
- CRC_violation
  This parameter indicates a corrupted AAL-PDU was detected due to an incorrect CRC result. A value of true in this parameter indicates the occurrence of this error.
- CRC_received
  This parameter indicates the value of the received CRC field when the CRC_violation parameter is set to true. This parameter is 4 bytes in length and can take on any value
- CRC_result

This parameter indicates the result of the CRC calculation at the receiver.

- Length_violation
  This parameter indicates a corrupted AAL-PDU was detected because of a discrepancy between the amount of data reassembled and the value encoded in the length field. A value of true in this parameter indicates the occurrence of this error.
- Length_field
  This parameter indicates the value received in the length field of the AAL-PDU.
- Invalid_format
  This parameter indicates a corrupted AAL-PDU was detected due to a non-zero Control field. A value of true in this parameter indicates the occurrence of this error.
- Reassembly_time_out
  This parameter indicates a reassembly error due to the expiration of the reassembly timer T1. A value of true in this parameter indicates the occurrence of this error.
- Oversized_received_sdu
  This parameter indicates an error due to a reception of a partial or whole AAL-PDU that contains an AAL-SDU exceeding the maximum allowed length. A value of true in this parameter indicates the occurrence of this error.
- Oversized_submitted_sdu
  This parameter indicates a service mis-usage error due to an AAL-SDU submitted from the higher layer with a length greater than the maximum SDU size allowed. A value of true in this parameter indicates the occurrence of this error.
- AAL_CEI
  This parameter indicates the value of the AAL Connection Endpoint Identifier.
- ATM_CEI
  This parameter indicates the value of the ATM Connection Endpoint Identifier.
- Mmax_sdu_send_length
  This parameter indicates the maximum allowed length, in bytes, of an AAL-SDU to be transmitted. This parameter can take on any integer value from 1 to 65535.
- Mmax_sdu_delivered_length
  This parameter indicates the maximum allowed length, in bytes, of an AAL-SDU to be delivered to the AAL user. This parameter can take on any integer value from 1 to 65535.
- MT1
  This parameter indicates the minimum time that a receiver must wait before it considers the transmission of a partially reassembled AAL-PDU overdue and can then terminate the reassembly. Acceptable values for this parameter are **for further study**.
- Mdeliver_errored
  This parameter indicates whether the AAL entity should invoke the optional errored delivery service. A value of true indicates the option should be invoked.

5.2 Connection Parameters

Every AAL-connection is expected to have connection parameters. They are:

- max_sdu_send_length
  This parameter indicates the maximum size SDU, in bytes, that may be sent over a given connection. At a transmitter, the value of this parameter is compared to the length of each AAL-SDU before it is transmitted. Any AAL-SDU that has a length greater than max_sdu_send_length is discarded and the event is reported to layer management. This parameter can take on any integer value from 1 to 65535.
- max_sdu_deliver_length
  This parameter indicates the maximum size SDU, in bytes, that may be delivered to an AAL user. At a receiver, the value of this parameter is compared to the length of each AAL-SDU before it is delivered. Any AAL-SDUs that has a length greater than max_sdu_deliver_length is discarded and the event is reported to layer management. As an option, the oversized AAL-SDU may be delivered to the AAL user with an indication that the AAL-SDU is oversized. This parameter can take on any integer value from 1 to 65535.
- T1
  This parameter is used to detect the loss of the last segment of a multi-segment AAL-PDU. The value of this parameter identifies the minimum length of time that a receiver will wait for the last segment after it has received the first segment of an AAL-PDU. The timer is started upon the reception of the first segment of a multi-segment AAL-PDU, and is stopped upon the reception of the last segment of a multi-segment AAL-PDU. The expiration of this timer indicates that the last segment was delayed. This parameter is set by layer management. Acceptable values for this parameter are **for further study**. The implementation of this timer is considered optional; however, this timer may be desirable in some implementations to support more efficient use of local resources.
- deliver_errored
  This parameter indicates whether the errored delivery option is invoked for this connection. A value of true indicates that errored data is to be delivered for this connection.

5.3 Services Expected from the ATM Layer

This AAL expects the ATM layer to provide the multiplexing and transport of 48 byte segments between communicating AAL-entities. The following primitives are used at the ATM SAP:

Primitives	Request	Indicate	Response	Confirm
ATM-DATA	X	X		

Table 20. ATM-SAP Services Required by Data AAL

The AAL-entity passes one segment, a loss-priority, and an SDU-type per ATM-DATA.request. In addition, the AAL-entity accepts one segment, any of two different values for SDU-type, and the congestion-experienced indication per ATM-DATA.indicate.

5.4 Functions

The following functions are performed by the Data AAL:

- AAL-PDU Generation
  This transmitter function appends the appropriate amount of padding to the AAL-SDU to make the entire AAL-PDU (including the control field, length and CRC-32) an integer multiple of 48 bytes, and adds the control field and the length field.
- CRC Generation
  This transmitter function computes the CRC (CRC-32) over the entire AAL-PDU. The CRC result is computed as described in the FDDI standards [15] [16]. This computation assumes that the remainder of the polynomial division is initialized to all ones and uses the following generator polynomial. The result of the CRC calcu-

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

  lation is placed in the CRC field.
- Segmentation
  This function, performed at the transmitter, removes successive 48 byte segments from the AAL-PDU beginning with the most significant byte and submits them to the ATM layer for transmission. The most significant byte of the remaining part of the AAL-PDU is considered the most significant byte of the next segment. All segments of an AAL-PDU except the last one are submitted to the ATM layer with the ATM SDU-type parameter equal to zero. The last segment of an AAL-PDU submitted to the ATM layer will have the ATM SDU-type parameter equal to one. The last segment of the AAL-PDU always contains the AAL-trailer.
- Reassembly
  This function, performed at the receiver, reconstructs the AAL-PDU from the segments received from the ATM layer by appending successive segments to the partial AAL-PDU. The last segment of an AAL-PDU is identified by an ATM SDU-type parameter equal to one when received in the ATM-DATA.indication primitive. After the AAL receives a segment with an SDU-type parameter equal to one, the next segment received is assumed to be the first segment of the subsequent AAL-PDU.
- CRC Validation
  This function, performed at the receiver, computes the CRC-32 over the entire

AAL-PDU using the algorithm described in the FDDI standards [15][16]. This computation assumes that the remainder of the polynomial division is initialized to all ones and that the same G(x) is used. If the CRC detects no errors in the AAL-PDU the remainder of the CRC calculation will be:

$$V(x) = x^{31} + x^{30} + x^{26} + x^{25} + x^{24} + x^{18} + x^{15} + x^{14} + x^{12} + x^{11} + x^{10} + x^{8} + x^{6} + x^{5} + x^{4} + x^{3} + x + 1$$

- AAL-SDU Recovery
  This function, performed at the receiver, uses the length field to check for cell loss or gain and to identify the AAL-SDU boundaries within the AAL-PDU. If no cell loss or gain is detected the AAL-SDU is extracted from the AAL-PDU. AAL-PDUs are identified as errored if the following inequality, where *n_segments* is the number of segments that were used in the reassembly of the AAL-PDU, is not satisfied:

$$0 \le (n\_segments \times 48 - length - 8) \le 47$$

5.5 AAL–PDU Structure and Encoding

5.5.1 PDU Structure

The AAL-PDU is illustrated in Figure 13, “AAL–PDU Format”

The AAL-PDU contains the following fields:

- User data
  This field contains the AAL-SDU. This field is byte aligned and can range from 0 to 65535 bytes in length.
- Pad
  This field is used to align the AAL-PDU on 48 byte boundaries. This field is byte aligned and can range from 0 to 47 bytes in length.
- Control
  This 2 byte field is reserved for supporting future AAL functions. This field is coded as all zeros.
- Length
  This 2 byte field indicates the length, in bytes, of the User Data field. This field can take on all integer values from 0 to 65535. However, for values in excess of 11454 bytes, the Hamming distance of the CRC drops from 4 to 3.
- CRC-32
  This 4 byte field contains the result of the CRC-32 calculation. This field can take any 32-bit value.

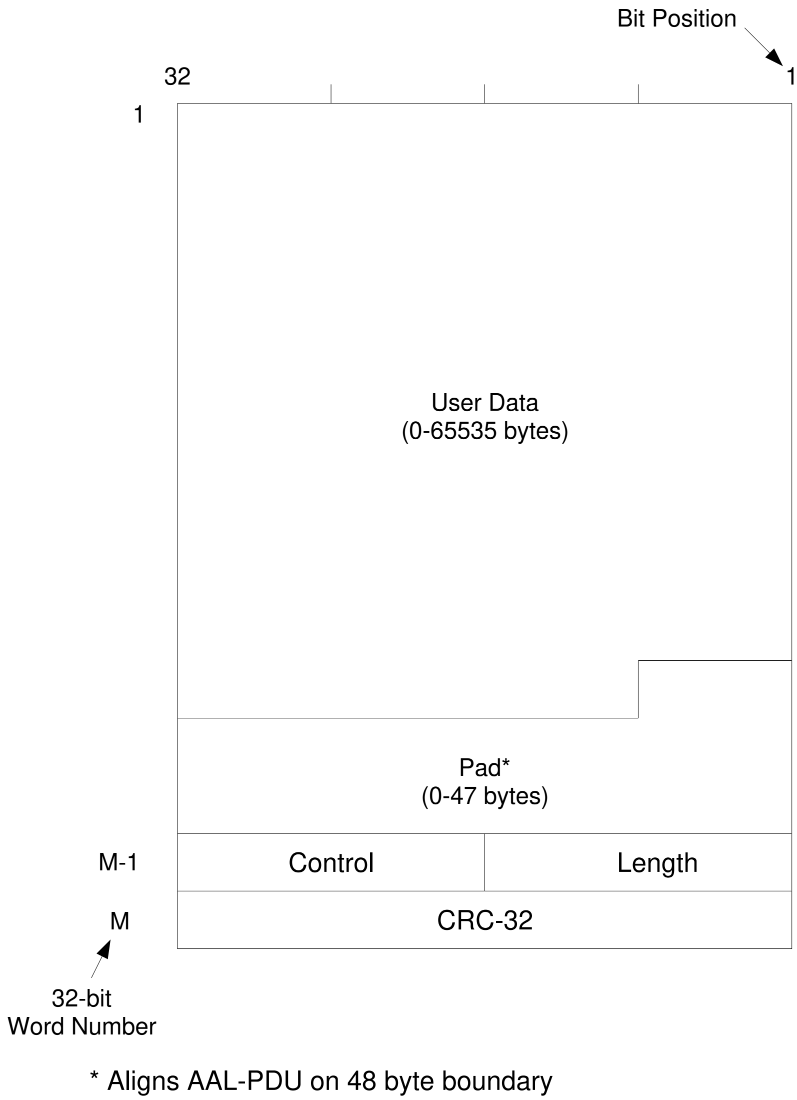


Figure 13. AAL–PDU Format

5.5.2 Encoding Principles

A field is encoded with its MSB in the highest number bit of the lowest number byte that the field spans. The remaining bits of the field, in progressively decreasing significance, are placed in decreasing bit positions within the same byte, then in increasing bytes. These encoding conventions are illustrated in Figure 14, "Data AAL Field Encoding Convention".

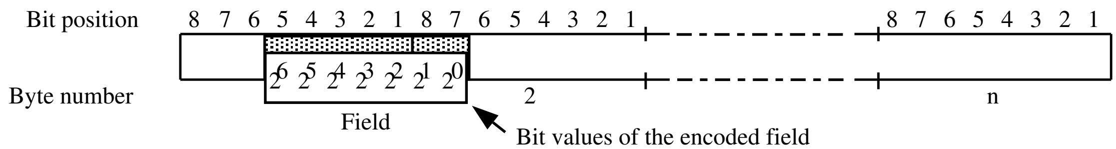


Figure 14. Data AAL Field Encoding Convention

5.6 AAL Procedures

The procedures describe the operation of the Data AAL protocol. This description of the AAL consists of two independent parts, one corresponding to the transmission of AAL-SDUs, and one corresponding to the reception of AAL-SDUs. These parts are referred to as the transmitter and the receiver, respectively.

This description does not include the errored data delivery option. The procedures describing errored data delivery are **for further study**.

5.6.1 Elements of Procedures

The following state variables are used within the procedures to illustrate the operation of the AAL:

- n_segments
  This variable represents the current number of segments used to reassemble the AAL-PDU. This variable is used to detect reassembly errors where the number of segments received is larger than the maximum number expected (max_number_-segments).
- max_number_segments
  This variable indicates the maximum number of segments that will be reassembled into an AAL-PDU for a given connection. At a receiver, this value is compared to the current number of segments received for a given AAL-PDU. If the number of segments received is equal to this value and the AAL-PDU is not complete, the PDU is discarded and the error is reported to system management. This parameter can take on any integer value from 1 to 1366. Implementations are not required to use max_number_segments or to verify on a per segment basis that the AAL-PDU is not too large; however, this feature may be desirable in some implementations to support more efficient use of local resources.
- r_crc
  This variable represents the result of the CRC calculation over the AAL-PDU at the receiver. The result of this variable is compared to the expected CRC result V(x)

and is used to detect corrupted AAL-PDUs.

5.6.2 Procedure Description for the Data AAL Transmitter

The transmitter state machine is assumed to process all request primitives according to a priority mechanism. There are two priorities of request, high and normal. The MAAL-RESET-TXR and MAAL-SET-NOW-TXR primitives are considered high priority requests. The MAAL-SET-TXR and AAL-UNITDATA request primitives are considered normal priority requests. The high priority requests are served sequentially, according to order of arrival, and are all served before any normal request primitives. The normal request primitives are served sequentially, according to order of arrival, and after all high priority request primitives have been served.

The State and Event description for the transmitter is summarized in Figure 15, "Data AAL Transmitter State Machine".

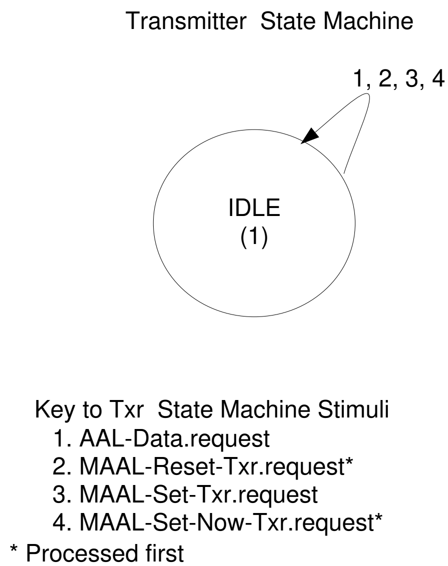


Figure 15. Data AAL Transmitter State Machine

(ER-74) Transmitting AAL-entities shall service all MAAL-RESET-TXR.request and MAAL-SET-NOW-TXR.request primitives sequentially according to their order of arrival and before all AAL-UNITDATA.request and MAAL-SET-TXR.request primitives. Transmitting AAL-entities shall service all AAL-UNIT-

DATA.request and MAAL-SET-TXR.request primitives sequentially according to their order of arrival.

(ER-75) Upon reception of an AAL-UNITDATA.request (transition[1] 1.1), if the AAL-SDU length is greater than max_sdu_send_length then the AAL-SDU is discarded, an MAAL-ERROR.indication is generated with only the oversized_submitted_sdu parameter set to true, and the transmitter returns to the idle state.

If the AAL-SDU length is less than or equal to max_sdu_length then the AAL-PDU is constructed as described in Section 5.5. Beginning with the most significant bit of the AAL-PDU successive 48 byte segments are removed and transferred to the ATM layer in ATM-DATA.request primitives with the SDU_type equal to zero until 48 bytes of the AAL-PDU remain. If the loss_priority parameter is high, these ATM-DATA.requests are issued with the ATM parameter loss-priority set to high. If the loss_priority parameter is set to normal or low, these ATM-DATA.requests are issued with the ATM parameter loss_priority set to low. The last 48 byte segment of the AAL-PDU is transferred to the ATM layer in an ATM-DATA.request primitive with the SDU-type parameter equal to 1. If the loss_priority parameter is high or normal, this ATM-DATA.request is issued with the ATM parameter loss-priority set to high. If the loss_priority parameter is set to low, this ATM-DATA.request is issued with the ATM parameter loss_priority set to low. The transmitter returns to the idle state.

(ER-76) Upon reception of an MAAL-RESET-TXR.request (transition 1.2), any remaining partially transmitted AAL-PDU is discarded along with any pending AAL-UNITDATA.requests. An MAAL-RESET-TXR.confirm is issued, and the transmitter returns to the idle state.

---

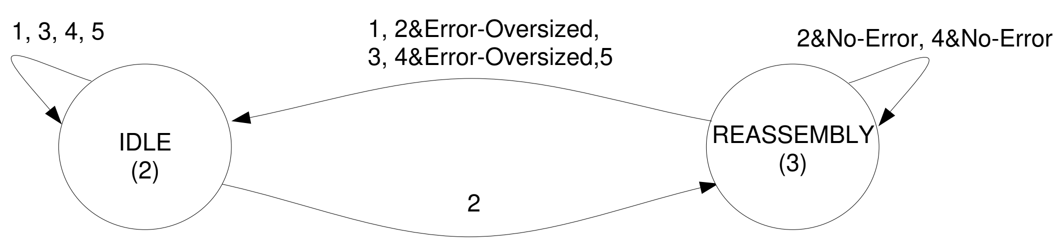
1. Transitions are identified by starting state number and stimulus number.

(ER-77) Upon reception of an MAAL-SET-TXR.request (transition 1.3), max_sdu_send_length is assigned the value of Mmax_sdu_send_length and the transmitter returns to idle.

(ER-78) Upon reception of an MAAL-SET-NOW-TXR.request (transition 1.4), max_sdu_send_length is assigned the value of Mmax_sdu_send_length and the transmitter returns to idle.

5.6.3 Procedure Description for the Data AAL Receiver

The State and Event description for the receiver is summarized in Figure 16, “Data AAL Receiver State Machine”.



Key to Receiver State Machine Stimuli
1. ATM-Data.indication(SDU-type = 1) (EOM)
2. ATM-DATA.indication(SDU-type = 0) (Not EOM)
3. Expiration of Reassembly Timer (T1)
4. MAAL-Set-Rcvr.request
5. MAAL-Reset-Rcvr.request

Figure 16. Data AAL Receiver State Machine

(EO-6.1) Updates to the optional timer T1 via an MAAL-SET-RCVR.request or an MAAL-CREATE.request shall apply to the next and all subsequent AAL-PDUs to initiate reassembly. Application of the new T1 value to any partially reassembled AAL-PDU is an implementation option.

(ER-79) Upon reception of an ATM.Data.indication with an SDU-type equal to 1 (EOM) and while in the idle state (transition 2.1), compute r_crc. If r_crc does not equal V(x), then the AAL-PDU is discarded, an MAAL-ERROR.indication is generated with only the CRC_violation parameter equal to true, and the receiver remains in the idle state.

Otherwise the control field is compared to all zeros, the value of the length field is compared to max_sdu_deliver_length, and cell loss or gain is determined by evaluating the following inequality:

$$0 \leq (48 - \text{length} - 8) \leq 47$$

If the inequality is not satisfied, or the control field does not equal zeros, or the length is greater than max_sdu_deliver_-length, then the AAL-PDU is discarded, an MAAL-ERROR.-indication is generated, and the receiver remains in the idle state. All parameters in a generated MAAL-ERROR.indication primitive except the length_violation, format_violation, and oversized_received_sdu are set to false. The length_violation would be set to true if the inequality was not satisfied, the format violation would be set to true if the control field was not all zeros, and the oversized_received_sdu would be set to true if length is greater than max_sdu_deliver_length.

Otherwise the AAL-SDU is extracted from the AAL-PDU and passed to the upper layer using an AAL-UNITDATA.indicate primitive and the receiver remains in the idle state.

(ER-80) Upon reception of an ATM.Data.indication with an SDU-type equal 0 (Not EOM) and while in the idle state (transition 2.2), the data is buffered, n_segments is set to 1, T1 is started, and the receiver goes to the reassembly state.

(EO-7) Upon expiration of T1 while in the idle state (transition 2.3), the receiver remains in the idle state.

(ER-81) Upon reception of an MAAL-SET-RCVR.request while in the idle state (transition 2.4), deliver_errored is assigned the value of Mdelivered_errored, T1 is assigned the value of MT1, max_-sdu_deliver_length is assigned the value of Mmax_sdu_deliv-er_length, max_number_segments is assigned as:

$$\text{max\_number\_segments} = \left\lceil \frac{(\text{max\_sdu\_deliver\_length} + 8)}{48} \right\rceil$$

where the ceiling function, $\lceil \; \rceil$, returns the next largest integer. The receiver remains in the idle state.

(ER-82) Upon reception of an MAAL-RESET-RCVR.request primitive while in the idle state (transition 2.5), timer T1 is stopped, MAAL-RESET-RCVR.confirm is issued, and the receiver remains in the idle state.

(ER-83) Upon reception of an ATM-DATA.indication with SDU-type = 1 (EOM) while in the reassembly state (transition 3.1), append data to the partial AAL-PDU, increment n_segments by one, stop timer T1, and compute r_crc. If r_crc does not equal V(x), then the AAL-PDU is discarded, an MAAL-ERROR.indication is generated with only the CRC_violation parameter equal to true, and the receiver goes to the idle state.

Otherwise the control field is compared to all zeros, the value of the length field is compared to max_sdu_deliver_length, and cell loss or gain is determined by evaluating following inequality:

$$0 \leq (\text{n\_segments} \times 48 - \text{length} - 8) \leq 47$$

If the inequality is not satisfied, or the control field does not equal zeros, or length is greater than max_sdu_deliver_length, then the AAL-PDU is discarded, an MAAL-ERROR.indication is generated, and the receiver goes to the idle state. All parameters in a generated MAAL-ERROR.indication primitive except the length_violation, format_violation, and oversized_received_sdu are set to false. The length_violation would be set to true if the inequality was not satisfied, the format violation would be set to true if the control field was not all zeros, and the oversized_received_sdu would be set to true if length is greater than max_sdu_deliver_length.

Otherwise the AAL-SDU is extracted from the AAL-PDU and passed to the upper layer using the AAL-UNITDATA.indicate primitive and the receiver goes to the idle state.

(ER-84) Upon reception of an ATM-DATA.indicate with SDU-type = 0 (Not EOM) while in the reassembly state (transition 3.2), append data to partial the AAL-PDU, increment n_segments by one. If n_segments equals max_number_segments then the partial AAL-PDU is discarded, T1 is stopped, an MAAL-ERROR.-indication is generated with only oversized_received_sdu set to true, and the receiver goes to the idle state. Otherwise, the receiver remains in the reassembly state.

(EO-8) Upon expiration of T1 while in the reassembly state (transition 3.3), the partial AAL-PDU is discarded, an MAAL-ERROR is generated with only the reassembly_time_out parameter set to true, and the receiver returns to the idle state.

(ER-85) Upon reception of an MAAL-SET-RCVR.request while in the reassembly state (transition 3.4), deliver_errored is assigned the value of Mdeliver_errored, T1 is assigned the value of MT1, max_sdu_deliver_length is assigned the value of Mmax_sdu_deliver_length, max_number_segments is assigned as:

$$\text{max\_number\_segments} = \left\lceil \frac{(\text{max\_sdu\_deliver\_length} + 8)}{48} \right\rceil$$

If n_segments is greater than or equal to max_number_segments then the partial AAL-PDU is discarded, T1 is stopped, an MAAL-ERROR.indication is generated with only oversized_received_sdu set to true, and the receiver goes to the idle state. Otherwise, the receiver remains in the reassembly state.

(ER-86) Upon reception of an MAAL-RESET-RCVR.request while in the reassembly state (transition 3.5), timer T1 is stopped, the partial AAL-PDU is discarded, MAAL-RESET-RCVR.confirm is issued, and the receiver goes to the idle state.

6. Permanent Virtual Circuit Managed Objects

The following defines an experimental portion of the Management Information Base (MIB) for use with network management protocols in TCP/IP-based internets. In particular, it defines objects for managing Permanent Virtual Circuit (PVC) parameters on Local ATM (LATM) switches.

The Internet-standard network management framework consists of: Structure and Identification of Management Information for TCP/IP-based internets, RFC 1155 [21], which describes how managed objects contained in the MIB are defined; Management Information Base for Network Management of TCP/IP-based internets, which describes the managed objects contained in the MIB, RFC 1156 [22]; and, the Simple Network Management Protocol, RFC 1157 [23], which defines the protocol used to manage these objects.

The SMI defines extensibility mechanisms. This document defines extensions to the MIB using the addition of widely-available but non-standard objects through the experimental subtree.

The LATM information store consists of several MIB modules, including the following, and, potentially, some other LATM and transmission MIB modules: MIB II [24] for general management of TCP/IP-based Internets, and LATM-PVC MIB (this chapter) for the management of LATM PVC parameters. Additional modules may be added in later phases and are **for further study**.

6.1 Objects

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the subset of Abstract Syntax Notation One (ASN.1) [25] defined in the "Structure and Identification of Management Information for TCP/IP-based internets" [21] (SMI). In particular, each object has a name, a syntax, and an encoding. The name is an object identifier, an administratively assigned name, which specifies an object type. The object type together with an object instance serves to uniquely identify a specific instantiation of the object. For human convenience, we often use a textual string, termed the OBJECT DESCRIPTOR, to also refer to the object type.

The syntax of an object type defines the abstract data structure corresponding to that object type. The ASN.1 language is used for this purpose. However, the SMI purposely restricts the ASN.1 constructs which may be used. These restrictions are explicitly made for implementation simplicity.

The encoding of an object type is simply how that object type is represented using the object type's syntax. Implicitly tied to the notion of an object type's syntax and encoding is how the object type is represented when being transmitted on the network. The SMI specifies the use of the basic encoding rules of ASN.1 [26], subject to the additional requirements imposed by SNMP.

6.1.1 Format of Definitions

Section 6.3 contains the specification of all object types contained in this MIB module. The object types are defined using the conventions defined in the SMI, as amended by the extensions specified in [27] [28].

6.2 Overview of the PVC MIB

Permanent Virtual Circuits are meant to be long-lived, established as the LATM switch is configured and the LATM network is brought up, then left in place. These PVCs may be uni- or bi-directional and may be uni- or multi-cast.

An SNMP manager will issue SNMP instructions (**get**, **get-next**, and **set**), to managed objects supported by the SNMP agent on the LATM switch. The SNMP agent will execute the instructions and respond (**get-response**). The managed objects available to the manager, and supported by the agent include the PVC MIB.

This PVC MIB is intended to provide the functionality necessary to set up, monitor and take down permanent virtual circuits on LATM switches.

6.2.1 Naming

In order to manipulate PVCs, the MIB must have a mechanism to identify a PVC and each of its members. PVC members would include every VPI/VCI virtual circuit of the PVC. A PVC member may connect a user to the LATM switch across the UNI, or interconnect two LATM switches across the local-NNI, or connect a LATM switch to the public ATM network. A PVC member is uniquely identified by the Switch, port, VPI and VCI; a PVC is the sum of its members. The PVC MIB also uses IDs as a shorthand for identifying PVCs and members. They are called PVCID and memberID respectively. The MIB includes two tables for specifying PVC members, one referenced by the port/VPI/VCI, the other referenced by the PVCID/memberID. Given either way of naming the PVC member, its parameters can be retrieved.

To identify a port on a LATM switch, the PVC MIB uses the technique of identifying each port with a unique integer (same value as ifIndex in MIB-II). A mapping between physically explicit names for ports and a logical identifier is then needed. The physically explicit names provide for identification of the switch, the chassis, the rack, the card, and the port. A unique identification of a port on a LATM switch is expected to be possible through this framework.A mapping must be provided in each direction: explicit to logical names, and logical to explicit names.

6.2.2 PVC Establishment

Setting up a PVC involves multiple steps, one for each member. The port associated with a member must be instructed to perform switching (input port to output port) and translation (input VPI/VCI to output VPI/VCI). The port may be unable to perform one or the other. For instance, a port may run out of VPI/VCI translation table memory, or there may be a collision in VPI/VCI name space. Before the PVC or any member is established, and cells are switched according to the port translation tables, all members must be verified as constructable. If any member step fails, the agent must notify the manager, which may abort the whole PVC set-up request, and reclaim the resources of each of the members. The PVC MIB uses a status object (with SYNTAX EntryStatus) to support this test/commit/ rollback functionality.

6.2.3 Configuration and Map Objects

Objects are defined in the configuration subtree to present the switch size and VPI/VCI ranges. They are inPortNumber, outPortNumber, VPIMask and VCIMask.

Two tables are defined in the map subtree to translate between the logical name and explicit name for a port. The first (logicalToExplicit) uses the logical name to reference the explicit names values. Each name's values can be retrieved separately, when "indexed" by the logical name. The second table (explicitToLogical) uses all the explicit names to reference the logical name's value. It can be retrieved when "indexed" by all the explicit names.

6.2.4 PVC Tables

Three tables are defined in the pvc subtree to count, create and edit PVCs and their members. The pvcCount table tallies the total number of PVCs established through a port.

The pvcSwitching table provides the functionality to setup and take down PVCs. A PVC is (conceptually) set up one member at a time. Creating a instance in the pvcSwitching table creates a member. To do this, the Status object's value must be set to createRequest. The Port, VPI and VCI must be specified also. (Because of the table "indexing," they are already specified indirectly, when referencing the Status objects.) If successful, the Status object's value will become underCreation. The agent will have assigned a PVCID and a memberID.

A member is added to a PVC in a fashion similar to the above, with the Status object's value set (createRequest), the Port, VPI, VCI specified indirectly. In addition the PVCID must be included to link this new member to the PVC.

When all members have been created and linked together, the PVC can be activated by setting the Status object's value to valid for one member, since the Status objects are linked. The PVC can be taken down by setting the Status object's value to invalid.

The pvc table provides the same information as the pvcSwitching table, plus more. It provides the functionality to edit PVCs. To edit a PVC, the Status object's value must first be set to underCreation. One feature of the pvc table is that its Status objects are linked with those of the pvcSwitching table, thus a PVC can be manipulated through either table. Through the pvc table, a member's Direction, BwNum and BwDenom objects can be set appropriately (default values are provided). A member's Port/VPI/VCI can be edited.

Because the pvc table is referenced by PVCID and memberID, an existing PVC member's Port, VPI and VCI can be determined from the PVCID and memberID.

6.3 Object Definitions

```
LATM-PVC-MIB DEFINITIONS ::= BEGIN

IMPORTS
    OBJECT-TYPE                          FROM RFC-1212
    TimeTicks                            FROM RFC1155-SMI;

--  This MIB module uses the extended OBJECT-TYPE macro as
--  defined in RFC1212

-- textual conventions
--
-- The VCI field of the ATM cell header.
--
VirtualCircuitIdentifier ::= INTEGER (0..65535)
--
-- The VPI field of the ATM cell header.
--
VirtualPathIdentifier ::= INTEGER (0..255)
--

-- The EntryStatus textual convention is also defined in RFC1271.
--
EntryStatus ::= INTEGER
```

```
    { valid(1),
    createRequest(2),
    underCreation(3),
    invalid(4)
    }
--
-- The status of a table entry.
--
-- Setting this object to the value invalid(4) has the
-- effect of invalidating the corresponding entry.
-- That is, it effectively disassociates the mapping
-- identified with said entry.
-- It is an implementation-specific matter as to whether
-- the agent removes an invalidated entry from the table.
-- Accordingly, management stations must be prepared to
-- receive tabular information from agents that corresponds
-- to entries currently not in use.  Proper
-- interpretation of such entries requires examination
-- of the relevant EntryStatus object.
--
-- An existing instance of this object cannot be set to
-- createRequest(2).  This object may only be set to
-- createRequest(2) when this instance is created.  When
-- this object is created, the agent may wish to create
-- supplemental object instances to complete a conceptual
-- row in this table.  Immediately after completing the
-- create operation, the agent must set this object to
-- underCreation(3).
--
-- Entries shall exist in the underCreation(3) state until
-- the management station is finished configuring the
-- entry and sets this object to valid(1) or aborts,
-- setting this object to invalid(4).
--
-- Requirements beyond RFC1271 follow:
-- Entries shall exist in the valid(1) state until the
-- management station wishes to edit the entry
-- and sets the object to underCreation(3),
-- or wishes to remove the entry by setting the object
-- to invalid(4).
--
-- Entries in a conceptual table may be linked to other entries
-- in the same or other tables, such that a change to one object
```

```
-- instance will be identically seen in all linked objects.
-- When such linkage exists, specific statements outlining its scope
-- will be made in the discription of the conceptual table.

--  This is the MIB module for the PVC related LATM switch objects.
--
latm     OBJECT IDENTIFIER ::= { experimental 31 }
switch   OBJECT IDENTIFIER ::= { latm 1 }
configurationOBJECT IDENTIFIER ::= { switch 1 }
map      OBJECT IDENTIFIER ::= { switch 2 }
pvc      OBJECT IDENTIFIER ::= { switch 3 }
--
-- The LATM Switch Input and Output Port count.
--
-- Basic LATM switch size information.

inPortNumber OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The total number of Input Ports on this LATM switch."
    ::= { configuration 1 }

OutPortNumber OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The total number of Output Ports on this LATM switch."
    ::= { configuration 2 }

VPIMask OBJECT-TYPE
    SYNTAX VirtualPathIdentifier
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "Identifies the range of VPI values supported by this switch.   The value is read as
        the unsigned integer value of a 8 bit mask where a 1 indicates this bit in the VPI  is
        supported, and a 0 indicates this bit is not supported."
    ::= { configuration 3 }

VCIMask OBJECT-TYPE
```

```
    SYNTAX VirtualCircuitIdentifier
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "Identifies the range of VCI values supported by this switch.   The value is read as
        the unsigned integer value of a 16 bit mask where a 1 indicates this bit in the VPI is
        supported, and a 0 indicates this bit is not supported."
    ::= { configuration 4 }

-- The Index Mapping Group
-- Implementation of the group is Mandatory.

-- Mapping Table from ifIndex to
-- Switch/Chassis/Rack/Card/Port
--
-- Allows a user to determine the explicit LATM switch interface
-- location, expressed by Switch/Chassis/Rack/Card/Port,
-- from the logical interface identifier, the ifIndex object.

logicalToExplicitTable  OBJECT-TYPE
    SYNTAX  SEQUENCE OF LogicalToExplicitEntry
    STATUS  mandatory
    DESCRIPTION
        This table contains mappings from the logical  interface identifier, the ifIndex ob-
        ject defined in RFC1213, to the local LATM switch identifiers for a port."
     ::= { map 1 }

logicalToExplicitEntry OBJECT-TYPE
    SYNTAX  LogicalToExplicitEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "This list contains ifIndex mapping parameters."
    INDEX   { logicalToExplicitIfIndex }
    ::= { logicalToExplicitTable 1 }

LogicalToExplicitEntry ::= SEQUENCE {
    logicalToExplicitIfIndex            INTEGER (1..65535),
    logicalToExplicitSwitch             INTEGER (1..65535),
    logicalToExplicitChassis            INTEGER (1..65535),
    logicalToExplicitRack               INTEGER (1..65535),
    logicalToExplicitCard               INTEGER (1..65535),
    logicalToExplicitPort               INTEGER (1..65535)
```

```
    }

logicalToExplicitIfIndexOBJECT-TYPE
    SYNTAX  INTEGER (1..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The value of this object identifies the LATM port interface for which this entry
        contains management information.  The value of this object for a particular inter-
        face has the same value as the ifIndex object, defined in RFC1213, for the same in-
        terface."
    ::= { logicalToExplicitEntry 1 }

logicalToExplicitSwitch  OBJECT-TYPE
    SYNTAX  INTEGER (1..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The value of this object identifies the switch for which this entry contains manage-
        ment information."
    ::= { logicalToExplicitEntry 2 }

logicalToExplicitChassis  OBJECT-TYPE
     SYNTAX  INTEGER (1..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The value of this object identifies the chassis for which this entry contains man-
        agement information."
    ::= { logicalToExplicitEntry 3 }

logicalToExplicitRack  OBJECT-TYPE
    SYNTAX  INTEGER (1..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The value of this object identifies the rack for which this entry contains manage-
        ment  information."
    ::= { logicalToExplicitEntry 4 }

logicalToExplicitCard  OBJECT-TYPE
    SYNTAX  INTEGER (1..65535)
    ACCESS  read-only
```

```
    STATUS  mandatory
    DESCRIPTION
        "The value of this object identifies the card for which this entry contains manage-
        ment information."
    ::= { logicalToExplicitEntry 5 }

logicalToExplicitPort  OBJECT-TYPE
    SYNTAX  INTEGER (1..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The value of this object identifies the port for which this entry contains manage-
        ment information."
    ::= { logicalToExplicitEntry 6 }

-- Mapping Table from Switch/Chassis/Rack/Card/Port
-- to ifIndex
--
-- Allows the user to determine the logical interface identifier,
-- the ifIndex object, corresponding to the explicit LATM switch
-- location — expressed as Switch/Chassis/Rack/Card/Port.

explicitToLogicalTable  OBJECT-TYPE
    SYNTAX  SEQUENCE OF ExplicitToLogicalEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "This table contains the mapping from the LATM switch particular identifiers for a
        port to the logical interface identifier, the ifIndex object."
    ::= { map 2 }

explicitToLogicalEntry OBJECT-TYPE
    SYNTAX  ExplicitToLogicalEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "This list contains ifIndex mapping parameters."
    INDEX   { explicitToLogicalSwitch,
        explicitToLogicalChassis,
        explicitToLogicalRack,
        explicitToLogicalCard,
        explicitToLogicalPort }
    ::= { explicitToLogicalTable 1 }
```

```
ExplicitToLogicalEntry ::= SEQUENCE {
    explicitToLogicalSwitch        INTEGER (1..65535),
    explicitToLogicalChassis       INTEGER (1..65535),
    explicitToLogicalRack          INTEGER (1..65535),
    explicitToLogicalCard          INTEGER (1..65535),
    explicitToLogicalPort          INTEGER (1..65535),
    explicitToLogicalIfIndex       INTEGER (1..65535)
    }

explicitToLogicalSwitch  OBJECT-TYPE
    SYNTAX  INTEGER (1..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The value of this object identifies the switch for which this entry contains manage-
        ment information."
    ::= { explicitToLogicalEntry 1 }

explicitToLogicalChassis  OBJECT-TYPE
    SYNTAX  INTEGER (1..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The value of this object identifies the chassis for which this entry contains man-
        agement information."
    ::= { explicitToLogicalEntry 2 }

explicitToLogicalRack  OBJECT-TYPE
    SYNTAX  INTEGER (1..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The value of this object identifies the rack for which this entry contains manage-
        ment information."
    ::= { explicitToLogicalEntry 3 }

explicitToLogicalCard  OBJECT-TYPE
    SYNTAX  INTEGER (1..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The value of this object identifies the card for which this entry contains manage-
```

```
        ment information."
    ::= { explicitToLogicalEntry 4 }

explicitToLogicalPort  OBJECT-TYPE
    SYNTAX  INTEGER (1..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The value of this object identifies the port for which this entry contains manage-
        ment information."
    ::= { explicitToLogicalEntry 5 }

explicitToLogicalIfIndexOBJECT-TYPE
    SYNTAX  INTEGER (1..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The value of this object identifies the LATM switch port interface for which this
        entry contains management information.  The value of  this object for a particular
        interface has the same value as the ifIndex object, defined in RFC1213, for the same
        interface."
    ::= { explicitToLogicalEntry 6 }


-- The Permanent Virtual Circuit Group
-- Implementation of the group is Mandatory.
--
-- LATM switch functionality involves both switching
-- and translation.  An incoming ATM cell is switched
-- through the switch fabric, from input to output port,
-- based on its incoming VPI and VCI.  In addition,
-- the VPI and VCI are translated by the switch.
--
-- Permanent Virtual Circuits (PVCs) are long-lived
-- switching and translation relationships.
-- They are established through and managed through
-- these tables.
--
-- A PVC may be multicast, or unicast.

-- The LATM Switch PVC Count Tables

pvcCountTable  OBJECT-TYPE
```

```
    SYNTAX  SEQUENCE OF PvcCountEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "This table enumerates the number of extant PVC connections, one entry per port."
    ::= { pvc 1 }

pvcCountEntry  OBJECT-TYPE
    SYNTAX  PvcCountEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "This list contains Port parameters."
    INDEX   { pvcCountIndex }
    ::= { pvcCountTable 1 }

PvcCountEntry  ::= SEQUENCE {
    pvcCountIndex  INTEGER (1..65535),
    pvcCountTotal  INTEGER (1..65535)
    }

pvcCountIndex  OBJECT-TYPE
    SYNTAX  INTEGER (1..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The value of this object identifies the port for which this entry contains manage-
        ment  information. The value of this object for a particular interface has the same
        value as the ifIndex object, defined in RFC1213, for the same interface."
    ::= { pvcCountEntry 1 }

pvcCountTotal OBJECT-TYPE
    SYNTAX INTEGER (1..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The total number of Permanent Virtual Circuits currently established through this
        port.  Both multicast and unicast PVCs are counted."
    ::= { pvcCountEntry 2 }

-- The LATM Switching and Translation Table
--
-- This table describes all PVCs on the switch.
```

```
--
-- This MIB table is writable.
-- It is expected that PVC setup and takedown
-- will all be executed through this table.
-- In particular, it is expected that PVC members will
-- be created through this table.

pvcSwitchingTable  OBJECT-TYPE
    SYNTAX  SEQUENCE OF PvcSwitchingEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "This table contains switching and translation parameters and state variables. A
        mapping  relates an input Port and input VPI/VCI with possibly multiple output
        Ports and output VPI/VCIs.  It defines both switching and translation functions.  It
        may define a bi-directional relation.  Such a mapping is named by an ID, and each
        constituent port/VPI/VCI is named by a sub-ID.  There will be multiple Switching
        Table entries for each such mapping, one entry for each port/VPI/VCI.  We identify
        a PVC with such a mapping, and a PVC member  with each port/VPI/VCI."
    ::= { pvc 2 }

pvcSwitchingEntry  OBJECT-TYPE
    SYNTAX  PvcSwitchingEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "This list contains Switching and Translation Table parameters and state variables."
    INDEX   { pvcSwitchingPort,
        pvcSwitchingVPI,
        pvcSwitchingVCI }
    ::= { pvcSwitchingTable 1 }

PvcSwitchingEntry  ::= SEQUENCE {
    pvcSwitchingPort                INTEGER (1..65535),
    pvcSwitchingVPI                 VirtualPathIdentifier,
    pvcSwitchingVCI                 VirtualCircuitIdentifier,
    pvcSwitchingPVCID               INTEGER (1..65535),
    pvcSwitchingMemberID            INTEGER (1..65535),
    pvcSwitchingCreationTime        TimeTicks,
    pvcSwitchingStatus              INTEGER
    }

pvcSwitchingPort  OBJECT-TYPE
```

```
    SYNTAX  INTEGER (1..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The value of this object identifies the port on which this entry's mapping is active.
        The value of this object for a particular interface has the same value as the ifIndex
        object, defined in RFC1213, for the same interface."
    ::= { pvcSwitchingEntry 1 }

pvcSwitchingVPI  OBJECT-TYPE
    SYNTAX  VirtualPathIdentifier
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The value of this object identifies the Virtual Path Identifier for which this entry's
        mapping is active."
    ::= { pvcSwitchingEntry 2 }

pvcSwitchingVCI  OBJECT-TYPE
    SYNTAX  VirtualCircuitIdentifier
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The value of this object identifies the Virtual Circuit Identifier for which this en-
        try's mapping is active."
    ::= { pvcSwitchingEntry 3 }

 pvcSwitchingPVCID  OBJECT-TYPE
    SYNTAX  INTEGER (1..65535)
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The value of this object uniquely identifies this Permanent Virtual Circuit.  It is set
        by the agent.  For a PVC, there are multiple Switching Table entries, which will all
        share a unique value."
    ::= { pvcSwitchingEntry 4 }

pvcSwitchingMemberID  OBJECT-TYPE
    SYNTAX  INTEGER (1..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The value of this object identifies members of a Permanent Virtual Circuit.  It  is
```

```
        set by the agent.  Each member of a PVC has a different MemberID."
    ::= { pvcSwitchingEntry 5 }

pvcSwitchingCreationTime  OBJECT-TYPE
    SYNTAX  TimeTicks
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The value of the sysUpTime object, as defined in  RFC-1213, at which this trans-
        lation entry was created."
    ::= { pvcSwitchingEntry 6 }

pvcSwitchingStatus OBJECT-TYPE
    SYNTAX  EntryStatus
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The status of this switching and translation entry.  Note that this object is linked to
        (its value is identical to) the values of other Status objects.  It mirrors other PVC
        member entries in this Switching Table, i.e., entries that share its PVCID.  It also
        mirrors entries in the pvc Table which share its PVCID."
    ::= { pvcSwitchingEntry 7 }



-- The LATM PVC Table
--
-- This table describes all PVC circuits on the switch,
-- and groups them by PVCID and memberID.
--
-- This MIB table is writable.
-- It is expected that PVC editing
-- will all be executed through this table.
--
--  Where such objects are common with the pvcSwitching table,
--  objects here have the same semantics as those in
-- the pvcSwitching Table.
--
-- Additional parameters are for further definition of a PVC member.

pvcTable  OBJECT-TYPE
    SYNTAX  SEQUENCE OF PvcEntry
    ACCESS  not-accessible
```

```
    STATUS  mandatory
    DESCRIPTION
        “This table contains Switching and Translation parameters and state variables.”
    ::= { pvc 3 }

pvcEntry  OBJECT-TYPE
    SYNTAX  PvcEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        “This list contains Switching and Translation parameters and state variables.”
    INDEX   { pvcPVCID,
        pvcMemberID }
    ::= { pvcTable 1 }

PvcEntry  ::= SEQUENCE {
    pvcPVCID                    INTEGER (1..65535),
    pvcMemberID                 INTEGER (1..65535),
    pvcPort                     INTEGER (1..65535),
    pvcVPI                      VirtualPathIdentifier,
    pvcVCI                      VirtualCircuitIdentifier,
    pvcDirection                INTEGER,
    pvcBwNum                    INTEGER (1..65535),
    pvcBwDenom                  INTEGER (1..65535),
    pvcStatus                   INTEGER
    }

pvcPVCID  OBJECT-TYPE
    SYNTAX  INTEGER (1..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        “The value of this object uniquely identifies this Permanent Virtual Circuit.  It is set
        by the agent.  For a PVC, there are multiple Switching Table entries, which will all
        share a unique value.”
    ::= { pvcEntry 1 }

pvcMemberID  OBJECT-TYPE
    SYNTAX  INTEGER (1..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        “The value of this object identifies members of a Permanent Virtual Circuit.  It  is
```

```
        set by the agent.  Each member of a PVC has a different MemberID."
    ::= { pvcEntry 2 }

pvcPort  OBJECT-TYPE
    SYNTAX  INTEGER (1..65535)
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The value of this object identifies the port on which this entry's mapping is active.
        The value of this object for a particular interface has the same value as the ifIndex
        object, defined in RFC1213, for the same interface."
    ::= { pvcEntry 3 }

pvcVPI  OBJECT-TYPE
    SYNTAX  VirtualPathIdentifier
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The value of this object identifies the Virtual Path Identifier for which this entry's
        mapping is active."
    ::= { pvcEntry 4 }

pvcVCI  OBJECT-TYPE
    SYNTAX  VirtualCircuitIdentifier
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The value of this object identifies the Virtual Circuit Identifier for which this en-
        try's mapping is active."
    ::= { pvcEntry 5 }

pvcDirection  OBJECT-TYPE
    SYNTAX  INTEGER {
        pvcDirectionOther (1),          -- unknown
        pvcDirectionRead (2),           -- user only receives across the UNI
        pvcDirectionWrite (3),          -- user only sends across the UNI
        pvcDirectionReadWrite (4)       -- user receives and sends across the UNI
    }
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The value of this object identifies whether this PVC member is uni-directional, or
        bi-directional."
```

```
    DEFVAL { 4 }
    ::= { pvcEntry 6 }

 pvcBwNum  OBJECT-TYPE
    SYNTAX  INTEGER (1..65535)
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The reserved bandwidth on the link is characterized by pvcSwitchingBwNum cell
        slots allocated over a time base of pvcSwitchingBwDenom cell slots.  These values
        are expressed independently of link speed."
    DEFVAL { 1 }
    ::= { pvcEntry 7 }

 pvcBwDenom  OBJECT-TYPE
    SYNTAX  INTEGER (1..65535)
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The reserved bandwidth on the link is characterized by pvcSwitchingBwNum cell
        slots allocated over a time base of pvcSwitchingBwDenom cell slots.  These values
        are expressed independently of link speed."
    DEFVAL { 1 }
    ::= { pvcEntry 8 }

pvcStatus OBJECT-TYPE
    SYNTAX  EntryStatus
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The status of this switching and translation entry.  Note that this object is linked to
        (its value is identical to) the values of other Status objects.  It mirrors other PVC
        member entries in this pvc Table, i.e., entries that share its PVCID.  It also mirrors
        entries in the Switching Table which share its PVCID."
    ::= { pvcEntry 9 }

END
```

7. ATM Layer Resource Management

One of the most important characteristics of an ATM network that distinguishes it from a traditional packet-switched network is that ATM provides the notion of bandwidth allocation that is absent in the latter. More specifically, in the ATM network, bandwidth allocation (i.e., reserving bandwidth) is possible through resource allocation within the network.

Bandwidth allocation within the network is extremely important to guarantee the QOS (Quality of Service) of multimedia applications. Among the most demanding multimedia applications are real time delivery of time-based information (e.g., video and audio) and non time-based information (e.g., image browsing). The former requires a minimum sustained bandwidth to maintain the intrinsic time relationships of the information contents to be delivered, while the latter requires bandwidth allocation to guarantee a maximum response time allowed for the application.

For example, to allocate a particular bandwidth (say B) for a VC, an ATM switch may be required to switch a fixed number of cells for this particular VC (say M) for every period of N time slots, where $B = C(M/N)$ and C = link bandwidth. This scheduling mechanism is equivalent to shaping a VC not to exceed a data rate of B. Examples of traffic shaping algorithms are sliding window algorithm, jumping window algorithm and leaky bucket mechanism.

Instead, for phase 1, a “virtual bandwidth allocation” mechanism will be implemented. This requires each workstation to implement some form of traffic shaping function to ensure its PVCs would not exceed a certain negotiated bandwidth. A table that contains all the existing PVCs and their allocated peak bandwidth must be maintained somewhere within the network. If a connection of a particular peak bandwidth needs to be established between two points in a network, this table needs to be consulted. If there exists enough bandwidth to support this new PVC, such connection would be granted and the table is updated. The bandwidth allocation within the network is observed by each workstation performing its own shaping.

In future phases, in order for applications to take advantage of the full capability of ATM, additional resource allocation functions (which include bandwidth and buffer allocation) to guarantee QOS may be required for the ATM switches.

8. Signalling in an ATM LAN

The topic of signalling, and more generally, of call control in general in a local ATM environment is very important to the long-term usability of this work. This work is intended to be included in phase 2 of this document and is expected to be based on Bellcore FA-NWT-1111[10].

9. Referenced Standards and Publications

This document is also intended to be used with the following standards and publications.

[1] 1990 CCITT Study Group XVIII Recommendation I.150, "B-ISDN Asynchronous Transfer Mode Functional Characteristics," CCITT, Geneva, 1990.

[2] 1990 CCITT Study Group XVIII Recommendation I.361, "B-ISDN ATM Layer Specification," CCITT, Geneva, 1990.

[3] 1990 CCITT Study Group XVIII Recommendation I.432, "B-ISDN User-Network Interface - Physical Layer Specification," CCITT, Geneva, 1990.

[4] CCITT Recommendation G.625.

[5] CCITT Recommendation G.957 (Draft) Optical Interfaces for Equipment and Systems Relating to the Synchronous Digital Hierarchy.

[6] ANSI T1.105-1988, "Digital Hierarchy — Optical Interface Rates and Formats."

[7] ANSI T1.ATM-199X Draft, "Broadband ISDN — ATM Layer Functionality and Specification," (T1S1.5/92-002R2).

[8] ANSI T1.INF-199X Draft, "Broadband ISDN — User-Network Interfaces, Rates and Formats Specification," (T1S1/91-634R1)

[9] ANSI Draft Standard, T1E1.2/92-020, "Broadband ISDN — Customer Installation Interfaces, Physical Media Dependent Specification," February 10, 1992.

[10] Bellcore Technical Reference TR-NWT-000253, "Synchronous Optical Network (SONET) Transport Systems: Common Generic Criteria," Bellcore Technical Reference, Issue 2, December 1991.

[11] Bellcore Technical Reference TR-TSY-000020, Issue 4.

[12] FDDI standard (ISO/IEC 9314-3).

[13] X. Widmer and P. A. Franaszek, "A DC-Balanced, Partitioned-Block, 8B/10B Transmission Code," *IBM Journal of Research and Development,* Vol. 27, No. 5: 440-451, September 1983.

[14] ANSI X3T9.3/92-007, "Fibre Channel, Physical and Signalling Interface (FC-PH), REV 2.2," January 24, 1992.

[15] ANSI X3.139-1987.

[16] ISO 9314-2:1989.

[17] T1S1.5/92-003R1, "Draft of the Broadband ISDN Standard on 'Variable Bit Rate Services ATM Adaptation Layer Unassured Data Transfer Protocol Functionality and Specification'," March 1992.

[18] CCITT Recommendation I.363, "B-ISDN ATM Adaptation Layer (AAL) Specification," Matsuyama, December 1991.

[19] CCITT TD 39 (Plenary) Appendices 9 and 10, "I.363 -- BISDN Adaptation Layer (AAL) Specification," Melbourne, December 1991.

[20] ANSI Committee T1 Contribution T1S1.5/91-449, "AAL 5 — A New High Speed Data Transfer AAL," IBM et al, November 1991, Dallas Texas.

[21] M.T. Rose and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets," Internet Working Group Request for Comments 1155. Network Information Center, SRI International, Menlo Park, California, (May 1990)

[22] K. McCloghrie and M.T. Rose, "Management Information Base for Network Management of TCP/IP-based Internets," Internet Working Group Request for Comments 1156. Network Information Center, SRI International, Menlo Park, California, (May 1990).

[23] J.D. Case, M.S. Fedor, M.L. Schoffstall, and J.R. Davin, "Simple Network Management Protocol," Internet Working Group Request for Comments 1157. Network Information Center, SRI International, Menlo Park, California, (May 1990).

[24] M.T. Rose (editor), "Management Information Base for Network Management of TCP/IP-based Internets," Internet Working Group Request for Comments 1213. Network Information Center, SRI International, Menlo Park, California, (March 1991).

[25] Information processing systems — Open Systems Interconnection —A Specification of Abstract Syntax Notation One (ASN.1), International Organization for Standardization. International Standard 8824, (December 1987).

[26] Information processing systems — Open Systems Interconnection — Specification of Basic Encoding Rules for Abstract Notation One (ASN.1), Interna-

tional Organization for Standardization. International Standard 8825, (December 1987).

[27] M.T. Rose, K. McCloghrie (editors), “Towards Concise MIB Definitions,” Internet Working Group Request for Comments 1212. Network Information Center, SRI International, Menlo Park, California, (March 1991).

[28] M.T. Rose (editor), “A Convention for Defining Traps for use with the SNMP,” Internet Working Group Request for Comments 1215. Network Information Center, SRI International, Menlo Park, California, (March 1991).

[29] Bellcore Framework Advisory FA-NWT-001111, “Broadband ISDN Access Signalling Framework Generic Criteria for Class II Equipment,” Bellcore Framework Advisory, Issue 1, December 1991.