

Network Working Group  
Request for Comments: 1695  
Category: Standards Track

M. Ahmed  
K. Tesink  
Editors  
Bell Communications Research  
August 1994

Definitions of Managed Objects  
for ATM Management Version 8.0  
using SMIV2

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Table of Contents

1. Introduction .....	2
2. The SNMPv2 Network Management Framework .....	2
3. Object Definitions .....	2
4. ATM Terminology .....	3
4.1 VCL/VPL and VCC/VPC .....	3
4.2 PVC and SVC .....	5
4.3 Traffic Management Parameters .....	5
4.3.1 Traffic Policing and Traffic Shaping Parameters .....	5
4.3.2 Cell Loss Priority .....	6
4.3.3 QoS Class .....	6
5. Overview .....	7
5.1 Background .....	7
5.2 Structure of the MIB .....	7
5.3 ATM Interface Configuration Group .....	7
5.4 ATM Interface DS3 PLCP and TC Layer Groups .....	8
5.5 ATM Virtual Link and Cross-Connect Groups .....	8
6. Application of MIB II to ATM .....	8
6.1 The System Group .....	8
6.2 The Interface Group .....	8
6.2.1 Support of the ATM Cell Layer by ifTable .....	9
7. Support of the AAL3/4 Based Interfaces .....	10
8. Support of the AAL5 Managed Objects .....	10
8.1 Managing AAL5 in a Switch .....	11
8.2 Managing AAL5 in a Host .....	12
8.3 Support of AAL5 by ifTable .....	13
8.4 Support of Proprietary Virtual Interface by ifT-able ..	14
8.5 AAL5 Connection Performance Statistics Group .....	15

9. ILMI MIB and the ATM Managed Objects .....	15
10. Definitions .....	18
11. Acknowledgments .....	72
12. References .....	72
13. Security Considerations .....	73
14. Authors' Addresses .....	73

## 1. Introduction

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes objects used for managing ATM-based interfaces, devices, networks and services.

This memo specifies a MIB module in a manner that is both compliant to the SNMPv2 SMI, and semantically identical to the peer SNMPv1 definitions.

## 2. The SNMPv2 Network Management Framework

The SNMPv2 Network Management Framework consists of four major components. They are:

- 0 RFC 1442 [1] which defines the SMI, the mechanisms used for describing and naming objects for the purpose of management.
- 0 STD 17, RFC 1213 [2] defines MIB-II, the core set of managed objects for the Internet suite of protocols.
- 0 RFC 1445 [3] which defines the administrative and other architectural aspects of the framework.
- 0 RFC 1448 [4] which defines the protocol used for network access to managed objects.

The Framework permits new objects to be defined for the purpose of experimentation and evaluation.

## 3. Object Definitions

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) defined in the SMI. In particular, each object type is named by an OBJECT IDENTIFIER, an administratively assigned name. 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 descriptor, to also refer to the object type.

4. ATM Terminology

Some basic ATM terminologies are described in this section to facilitate defining the ATM managed objects.

4.1. VCL/VPL and VCC/VPC

There are two distinct types of ATM virtual connections: Virtual Channel Connections (VCCs) and Virtual Path Connection (VPCs). As shown in Figures 1 and 2, ATM virtual connections consist of concatenated series of virtual links which forms a path between two end points, with each concatenation occurring at an ATM switch. Virtual links of VCCs are called Virtual Channel Links (VCLs). Virtual links of VPCs are called Virtual Path Links (VPLs). The VCI and VPI fields in the ATM cell header associate each cell of a VCC with a particular VCL over a given physical link. The VPI field in the ATM cell header associates each cell of a VPC with a particular VPL over a given physical link. Switches route cells between VCLs (or VPLs) via a cross-connect function according to the cells' VCI/VPI (or VPI) values.

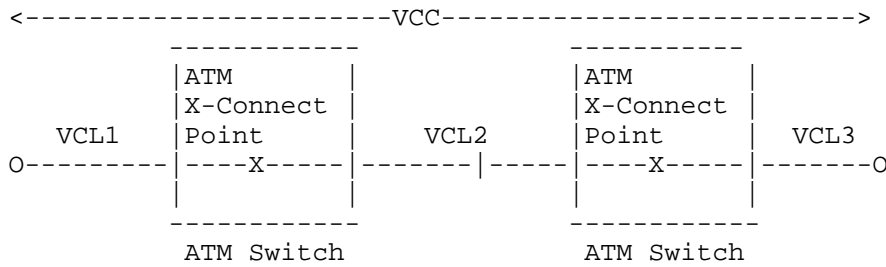


Figure 1: Virtual Channel Links and Virtual Channel Connection

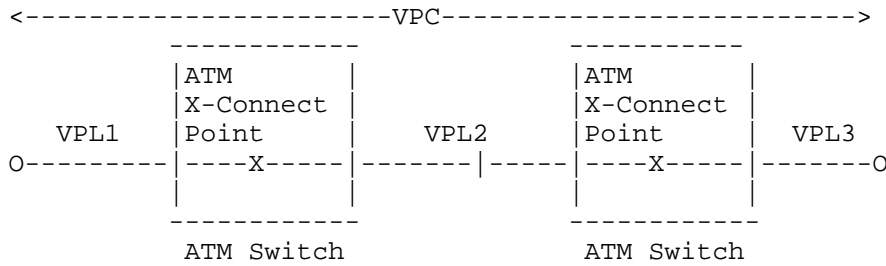


Figure 2: Virtual Path Links and Virtual Path Connection

A single ATM end-system or switch does not support the whole end-to-end span of a VCC (or VPC). Rather, multiple ATM end-systems and/or switches each support one piece of the VCC (or VPC). That is, each ATM end-system at one end of the VCC/VPC supports its end of the VCC/VPC plus the VCLs or VPLs on its external interfaces, and each switch through which the VCC/VPC passes, supports the multiple VCLs/VPLs on that switch's external interfaces and the cross-connection of those VCLs/VPLs through that switch. Thus, the end-to-end management of a VCC or VPC is achieved only by appropriate management of its individual pieces in combination.

Note that for management purposes, an ATM network may be viewed as a large distributed switch by hiding all the network's internal connectivity as being internal to the distributed switch (as shown in Figure 2a). This model may for example be used for Customer Network Management (CNM) purposes.

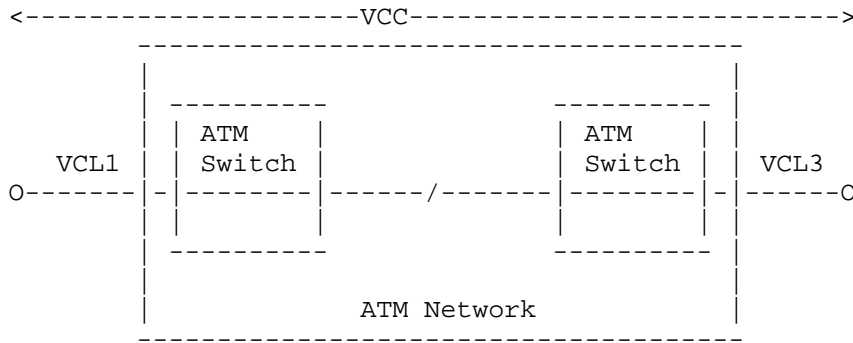


Figure 2a: ATM Network modeled as a large distributed switch

A VCC has a set of traffic characteristics (i.e., bandwidth parameters, QoS Class parameters, etc.). VCLs inherit their traffic characteristics from the VCC of which they are a part. VCCs are bi-directional by definition. However, the traffic parameters in the two directions of a connection can be symmetric or asymmetric, i.e., the two directions can have the same or different traffic flows. A uni-directional traffic flow across a VCC is achieved by assigning a zero bandwidth in one direction. Note that in addition to the bandwidth required by the user traffic flow, bandwidth is also required for OAM cell flows, even for the zero-bandwidth direction of a uni-directional connection. These same principles apply to VPCs.

#### 4.2. PVC and SVC

A Permanent Virtual Connection (PVC) is a provisioned VCC or VPC. A Switched Virtual Connection (SVC) is a switched VCC or VPC that is set up in real-time via call set-up signaling procedures. A PVC (or an SVC) can be a point-to-point, point-to-multipoint, or multipoint-to-multipoint VCC or VPC.

#### 4.3. Traffic Management Parameters

##### 4.3.1. Traffic Policing and Traffic Shaping Parameters

In order to allocate resources fairly among different users, some networks police traffic at resource access points. The traffic enforcement or policing taken at a UNI is called Usage Parameter Control (UPC) and is activated on an incoming VCL or VPL as shown in Figure 3. The use of the traffic enforcer at the ingress of the connection is to make sure that the user traffic does not exceed the

negotiated traffic parameters such as the peak cell rate associated with a specific traffic descriptor type.

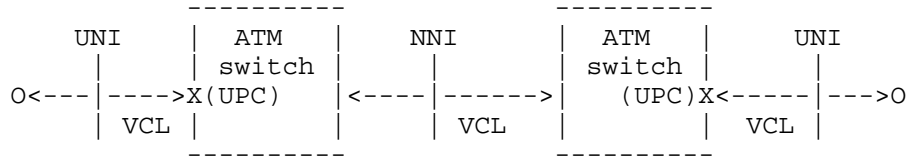


Figure 3: An Example of a UPC

In addition, traffic shaping may be performed on an outgoing VPL or VCL at a given ATM interface. The function of the ATM traffic shaper either at the source or an egress point of the connection is to smooth the outgoing cell traffic inter-arrival time. If policing or shaping is not performed then the policing or shaping algorithm is not activated. ATM Forum has specified seven traffic descriptor types including one for the best effort traffic [9].

#### 4.3.2. Cell Loss Priority

To prioritize traffic during resource congestion, ATM cells are assigned one of the two types of Cell Loss Priority (CLP), CLP=0 and CLP=1. ATM cells with CLP=0 have a higher priority in regard to cell loss than ATM cells with CLP=1. Therefore, during resource congestions, CLP=1 cells are dropped before any CLP=0 cell is dropped.

#### 4.3.3. QoS Class

A VCC or VPC is associated with one of a number of Quality of Service (QoS) classes. The following service classes have been specified:

- Service Class A: Constant bit rate video and Circuit emulation
- Service Class B: Variable bit rate video/audio
- Service Class C: Connection-oriented data
- Service Class D: Connectionless data

Four QoS classes numbered 1, 2, 3, and 4 have been specified with the aim of supporting service classes A, B, C, and D respectively. The VCLs (or VPLs) concatenated to form a VCC (or VPC) will all have the same QoS class as that of the VCC (or VPC). The Cell Loss Ratio (CLR), Cell Delay Variation (CDV), and end-to-end Cell Delay (CD) parameters are defined as part of QoS Class definition. In addition,

an unspecified QoS Class numbered 0 is specified for best effort traffic.

## 5. Overview

ATM management objects are used to manage ATM interfaces, ATM virtual links, ATM cross-connects, AAL5 entities and AAL5 connections supported by ATM hosts, ATM switches and ATM networks. This section provides an overview and background of how to use this MIB and other potential MIBs for this purpose.

The purpose of this memo is primarily to manage ATM PVCs. ATM SVCs are also represented by the management information in this MIB. However, full management of SVCs may require additional capabilities which are beyond the scope of this memo.

### 5.1. Background

In addition to the MIB module defined in this memo, other MIB modules are necessary to manage ATM interfaces, links and cross-connects. Examples include MIB II for general system and interface management (RFC 1213 and RFC 1573), the DS3 or SONET MIBs for management of physical interfaces, and, as appropriate, MIB modules for applications that make use of ATM, such as SMDS. These MIB modules are outside the scope of this specification.

The current specification of this ATM MIB is based on SNMPv2.

### 5.2. Structure of the MIB

The managed ATM objects are arranged into the following groups:

- (1) ATM interface configuration group
- (2) ATM interface DS3 PLCP group
- (3) ATM interface TC Sublayer group
- (4) ATM interface virtual link (VPL/VCL) configuration groups
- (5) ATM VP/VC cross-connect groups
- (6) AAL5 connection performance statistics group

Note that, managed objects for activation/deactivation of OAM cell flows and ATM traps notifying virtual connection or virtual link failures are outside the scope of this memo.

### 5.3. ATM Interface Configuration Group

This group contains information on ATM cell layer configuration of local ATM interfaces on an ATM device in addition to the information

on such interfaces contained in the ifTable.

#### 5.4. ATM Interface DS3 PLCP and TC Layer Groups

These groups provide performance statistics of the DS3 PLCP and TC sublayer of local ATM interfaces on a managed ATM device. DS3 PLCP and TC sublayer are currently used to carry ATM cells respectively over DS3 and SONET transmission paths.

#### 5.5. ATM Virtual Link and Cross-Connect Groups

ATM virtual link and cross-connect groups model bi-directional ATM virtual links and ATM cross-connects. The ATM VP/VC link groups are implemented in an ATM host, ATM switch and ATM network. The ATM switch and ATM network also implement the ATM VP/VC cross-connect groups. Both link and cross-connect groups are implemented in a carrier's network for Customer Network Management (CNM) purposes.

The ATM virtual link groups are used to create, delete or modify ATM virtual links in an ATM host, ATM switch and ATM network. ATM virtual link groups along with the cross-connect groups are used to create, delete or modify ATM cross-connects in an ATM switch or ATM network (e.g., for CNM purposes).

### 6. Application of MIB II to ATM

#### 6.1. The System Group

For the purposes of the sysServices object in the System Group of MIB II [2], ATM is a data link layer protocol. Thus, for ATM switches and ATM networks, sysServices will have the value "2".

#### 6.2. The Interface Group

The Interfaces Group of MIB II defines generic managed objects for managing interfaces. This memo contains the media-specific extensions to the Interfaces Group for managing ATM interfaces.

This memo assumes the interpretation of the Interfaces Group to be in accordance with [5] which states that the interfaces table (ifTable) contains information on the managed resource's interfaces and that each sub-layer below the internetwork layer of a network interface is considered an interface. Thus, the ATM cell layer interface is represented as an entry in the ifTable. This entry is concerned with the ATM cell layer as a whole, and not with individual virtual connections which are managed via the ATM-specific managed objects specified in this memo. The inter-relation of entries in the ifTable is defined by Interfaces Stack Group defined in [5].



## 6.2.1. Support of the ATM Cell Layer by ifTable

Some specific interpretations of ifTable for the ATM cell layer follow.

Object	Use for the generic ATM layer
=====	=====
ifIndex	Each ATM port is represented by an ifEntry.
ifDescr	Description of the ATM interface.
ifType	The value that is allocated for ATM is 37.
ifSpeed	The total bandwidth in bits per second for use by the ATM layer.
ifPhysAddress	The interface's address at the ATM protocol sublayer; the ATM address which would be used as the value of the Called Party Address Information Element (IE) of a signalling message for a connection which either: <ul style="list-style-type: none"> <li>- would terminate at this interface, or</li> <li>- for which the Called Party Address IE would need to be replaced by the Called Party SubAddress IE before the message was forwarded to any other interface.</li> </ul> For an interface on which signalling is not supported, then the interface does not necessarily have an address, but if it does, then ifPhysAddress is the address which would be used as above in the event that signalling were supported. If the interface has multiple such addresses, then ifPhysAddress is its primary address. If the interface has no addresses, then ifPhysAddress is an octet string of zero length. Address encoding is as per [9]. Note that addresses assigned for purposes other than those listed above (e.g., an address associated with the service provider side of a public network UNI) may be represented through atmInterfaceAdminAddress.
ifAdminStatus	See [5].
ifOperStatus	Assumes the value down(2) if the ATM cell layer or any layer below that layer is down.

ifLastChange See [5].

ifInOctets The number of received octets over the interface, i.e., the number of received, assigned cells multiplied by 53.

ifOutOctets The number of transmitted octets over the interface, i.e., the number of transmitted, assigned cells multiplied by 53.

ifInErrors The number of cells dropped due to uncorrectable HEC errors.

ifInUnknownProtos The number of received cells discarded during cell header validation, including cells with unrecognized VPI/VCI values, and cells with invalid cell header patterns. If cells with undefined PTI values are discarded, they are also counted here.

ifOutErrors See [5].

ifName Textual name (unique on this system) of the interface or an octet string of zero length.

ifLinkUpDownTrapEnable Default is disabled (2).

ifConnectorPresent Set to false (2).

ifPromiscuousMode Set to false(2).

ifHighSpeed See [5].

ifHCInOctets The 64-bit version of ifInOctets; supported if required by the compliance statements in [5].

ifHCOctets The 64-bit version of ifOutOctets; supported if required by the compliance statements in [5].

## 7. Support of the AAL3/4 Based Interfaces

For the management of AAL3/4 CPCS layer, see [6].

## 8. Support of the AAL5 Managed Objects

Support of AAL5 managed objects in an ATM switch and ATM host are described below.

## 8.1. Managing AAL5 in a Switch

Managing AAL5 in a switch involves:

- (1) performance management of an AAL5 entity as an internal resource in a switch
- (2) performance management of AAL5 per virtual connection

AAL5 in a switch is modeled as shown in Figures 4 and 5. AAL5 will be managed in a switch for only those virtual connections that carry AAL5 and are terminated at the AAL5 entity in the switch. Note that, the virtual channels within the ATM UNIs carrying AAL5 will be switched by the ATM switching fabric (termed as ATM Entity in the figure) to the virtual channels on a proprietary internal interface associated with the AAL5 process (termed as AAL5 Entity in the figure). Therefore, performance management of the AAL5 resource in the switch will be modeled using the ifTable through an internal (pseudo-ATM) virtual interface and the AAL5 performance management per virtual connection will be supported using an additional AAL5 connection table in the ATM MIB. The association between the AAL5 virtual link at the proprietary virtual, internal interface and the ATM virtual link at the ATM interface will be derived from the virtual channel cross-connect table and the virtual channel link table in the ATM MIB.

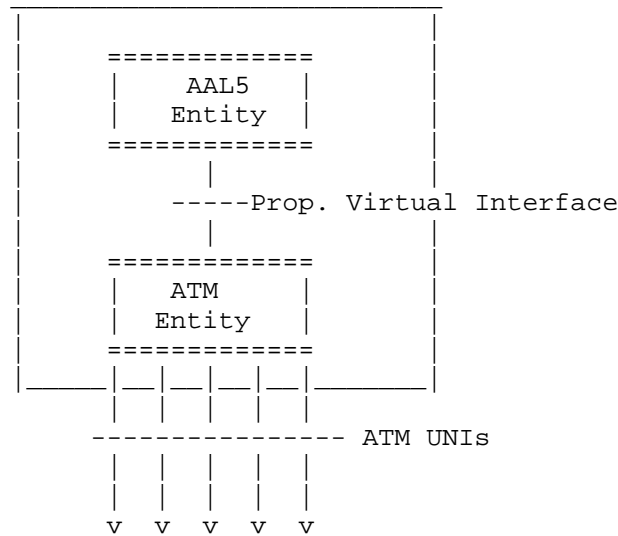


Figure 4 : Model of an AAL5 Entity in a Switch

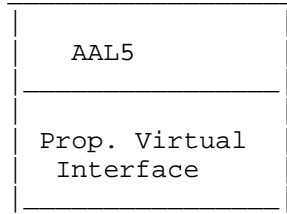


Figure 5 : AAL5 Entity's Interface Stack in a Switch

8.2. Managing AAL5 in a Host

Managing AAL5 in a host involves managing the AAL5 sublayer interface as shown in Figures 6 and 7. The AAL5 sublayer is stacked directly over the ATM sublayer. The ifTable is applied to the AAL5 sublayer as defined in Section 8.3.

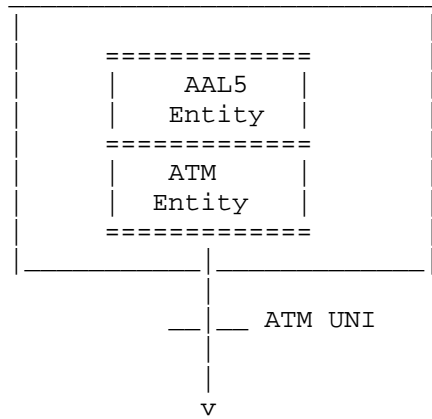


Figure 6 : Model of an AAL5 Entity in a Host

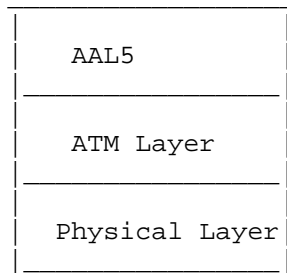


Figure 7 : AAL5 Entity's Interface Stack in a Host

## 8.3. Support of AAL5 by ifTable

The AAL5 entity in an ATM device (e.g., switch or host) is managed using the ifTable. There are additional counters specified for AAL5 than those specified in the ATM B-ICI document [10]. Specific interpretations of ifTable for the AAL5 CPCS layer are as follows.

Object	Use for AAL5 CPCS layer entity
=====	=====
ifIndex	Each AAL5 entity is represented by an ifEntry.
ifDescr	Description of the AAL5 entity.
ifType	The value that is allocated for AAL5 is 49.
ifMtu	Set to the largest PDU size for the AAL5 CPCS layer that can be processed by the AAL5 entity.
ifSpeed	Set to 0.
ifPhysAddress	An octet string of zero length.
ifAdminStatus	See [5].
ifOperStatus	Assumes the value down(2) if the AAL5 or any layer below that layer is down.
ifLastChange	See [5].
ifInOctets	The number of received AAL5 CPCS PDU octets.
ifOutOctets	The number of AAL5 CPCS PDU octets transmitted.
ifInUcastPkts	The number of received AAL5 CPCS PDUs passed to a higher-layer.
ifOutUcastPkts	The number of AAL5 CPCS PDUs received from a higher-layer for transmission. [Note: The number of AAL5 PDUs actually transmitted is the number received from a higher-layer for transmission minus any which are counted by ifOutErrors and ifOutDiscards.]

ifInErrors        Number of errored AAL5 CPCS PDUs received.  
                   The types of errors counted include CRC-32 errors,  
                   SAR time-out errors, and oversized SDU errors.

ifInUnknownProtos Set to 0.

ifInDiscards     Number of received AAL5 CPCS PDUs discarded.  
                   Possible reason may be input buffer overflow.

ifOutErrors      Number of AAL5 CPCS PDUs that could not  
                   be transmitted due to errors.

ifOutDiscards    Number of AAL5 CPCS PDUs received for  
                   transmission that are discarded.  
                   Possible reason may be output buffer  
                   overflow.

ifInMulticastPkts Set to 0.

ifInBroadcastPkts Set to 0.

ifOutMulticastPkts Set to 0.

ifOutBroadcastPkts Set to 0.

ifName        Textual name (unique on this system) of the  
                   AAL5 entity or an octet string of zero length.

ifHighSpeed        Set to 0.

ifConnectorPresent Set to false (2).

ifPromiscuousMode Set to false(2).

ifLinkUpDownTrapEnable        Default is disabled (2).

8.4. Support of Proprietary Virtual Interface by ifTable

Specific interpretations of ifTable for the proprietary virtual, internal interface associated with an AAL5 entity in an ATM switch are as follows.

Object	Use for proprietary virtual, internal interface associated with AAL entities
=====	=====
ifIndex	Each proprietary virtual, internal interface associated with AAL entities is represented by an

ifEntry.

ifDescr Description of the proprietary virtual, internal interface associated with AAL entities.

ifType The value that is allocated for proprietary virtual, internal interface is 53.

ifSpeed See [5]. Set to 0 if the speed is not known.

ifPhysAddress See [5]. An octet string of zero length if no address is used for this interface.

ifAdminStatus See [5].

ifOperStatus See [5].

ifLastChange See [5].

ifName Textual name (unique on this system) of the interface or an octet string of zero length.

ifHighSpeed See [5]. Set to 0 if the speed is not known.

ifConnectorPresent Set to false (2).

ifLinkUpDownTrapEnable Default is disabled (2).

#### 8.5. AAL5 Connection Performance Statistics Group

An AAL5 connection table is used to provide AAL5 performance information for each AAL5 virtual connection that is terminated at the AAL5 entity contained within an ATM switch or host.

#### 9. ILMI MIB and the ATM Managed Objects

The ILMI MIB is specified by the ATM Forum in UNI specification [9], to manage local ATM UNIs. The support of the ATM management functions by the ILMI MIB and those contained in this memo are compared in Table 1. In this table, "yes" in the "ILMI MIB" column indicates that the management functions are supported by the ILMI MIB. The MIB groups in the "This memo" column are the groups listed in Section 5.2.

For that subset of management information which the ILMI MIB and this memo have in common, every effort has been made to retain identical semantics and syntax, even though the MIB objects are identified

using different OBJECT IDENTIFIERS.

Table 1 - Structuring of ATM Managed Objects

ATM Mgmt.Inf.	ATM Managed Objects	This memo	ILMI MIB
---------------	---------------------	-----------	----------

Local Interface Information:

ATM interface: physical layer configuration	(1) port identifier (2) physical transmission types (3) operational status (4) administrative status (5) last change status	ATM MIB gr.1* MIB II	yes*
ATM interface: cell layer configuration	(1) active VPI/VCI fields (2) maximum number of VPCs/VCCs (3) configured VPCs/VCCs (4) ILMI VPI/VCI values (5) ATM address type (6) ATM administrative address	ATM MIB gr.1	yes **
ATM interface: cell layer performance	(1) received/transmitted cells (2) cells with HEC error (3) cell header validation errors	MIB II	yes
ATM interface: PLCP & TC layer performance	(1)DS3 PLCP severely errored framing seconds (2)DS3 PLCP unavailable seconds (3)DS3 PLCP alarm state (4)out of cell delineation events (5)TC alarm state	ATM MIB gr.2,3	no
VP/VC link: configuration	(1)VPI or VPI/VCI value (2)VCL or VPL operational status (3)VCL/VPL administrative status (4)VCL/VPL last change status (5)transmit/receive traffic/QoS parameters (6)AAL type (7)transmit/receive AAL5 SDU size (8)AAL5 encapsulation type	ATM MIB gr. 4	yes ***



VP/VC Cross-connect: configuration	(1)cross-connect identifier (2)port identifier of one end (3)port identifier of the other end (4)VPI or VPI/VCI value of one end (5)VPI or VPI/VCI value of the other end (6)VC/VP cross-connect operational status (7)VC/VP cross-connect administrative status (8)VC/VP last change status	ATM MIB gr. 5	no
VCC AAL5 CPCS layer: performance	(1)PDUs discarded for CRC errors (2)PDUs discarded due to reassembly time out (3)PDUs discarded due to large SDUs	ATM MIB gr.6	no
AAL5 entity:	(1)received/transmitted PDUs (2)PDUs discarded due to protocol errors (3)a set of configuration/state parameters	MIB II	no

\*The operational, administrative, and last change status of the ATM interface and the physical transmission type shall be supported by the interface table in MIB II (RFC 1213, RFC 1573). ILMI does not contain the administrative and last change status of the ATM interface.

\*\* The ILMI MIB does not contain information on the ATM address type and the ATM administrative address assigned at the ATM interface.

\*\*\*The ILMI MIB contains local and end-to-end operational status of the VPC/VCC segment. However, it does not contain the VPC/VCC administrative and last change status and the VCC AAL information.

## 10. Definitions

```

ATM-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, OBJECT-IDENTITY,
    Counter32, Integer32, IpAddress
        FROM SNMPv2-SMI
    TEXTUAL-CONVENTION, DisplayString,
    TimeStamp, RowStatus
        FROM SNMPv2-TC
    MODULE-COMPLIANCE, OBJECT-GROUP
        FROM SNMPv2-CONF
    ifIndex, mib-2
        FROM RFC1213-MIB;

atmMIB MODULE-IDENTITY
    LAST-UPDATED "9406072245Z"
    ORGANIZATION "IETF ATOM MIB Working Group"
    CONTACT-INFO
        "
            Masuma Ahmed
            Postal: Bellcore
                  331 Newman Springs Road
                  Red Bank, NJ 07701
                  US
            Tel:    +1 908 758 2515
            Fax:    +1 908 758 4131
            E-mail: mxa@mail.bellcore.com

            Kaj Tesink
            Postal: Bellcore
                  331 Newman Springs Road
                  Red Bank, NJ 07701
                  US
            Tel:    +1 908 758 5254
            Fax:    +1 908 758 4196
            E-mail: kaj@cc.bellcore.com"
    DESCRIPTION
        "This is the MIB Module for ATM and AAL5-related
        objects for managing ATM interfaces, ATM virtual
        links, ATM cross-connects, AAL5 entities, and
        and AAL5 connections."
    ::= { mib-2 37 }

atmMIBObjects OBJECT IDENTIFIER ::= {atmMIB 1}

-- This ATM MIB Module consists of the following groups:

```

```

-- (1) ATM Interface configuration group
-- (2) ATM Interface DS3 PLCP group
-- (3) ATM Interface TC Sublayer group
-- (4) ATM Interface VPL configuration group
-- (5) ATM Interface VCL configuration group
-- (6) ATM VP Cross Connect group
-- (7) ATM VC Cross Connect group
-- (8) ATM Interface AAL5 VCC performance statistics
--      group

IfIndex ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "The value of this object identifies the interface
         for which the entry contains management
         information. The value of this object for a
         particular interface has the same value as the
         ifIndex object, defined in RFC 1213, for the same
         interface."
    SYNTAX      Integer32

AtmTrafficDescrParamIndex ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "The value of this object identifies the row
         in the atmTrafficDescrParamTable."
    SYNTAX      Integer32

atmTrafficDescriptorTypes  OBJECT IDENTIFIER ::=
                                {atmMIBObjects 1}

-- The following values are defined for use as
-- possible values of the ATM traffic descriptor type.
-- ATM Forum specified seven types of ATM traffic
-- descriptors.

atmNoTrafficDescriptor  OBJECT-IDENTITY
    STATUS      current
    DESCRIPTION
        "This identifies the no ATM traffic descriptor
         type. Parameters 1, 2, 3, 4, and 5 are not used.
         This traffic descriptor type can be used for
         best effort traffic."
    ::= { atmTrafficDescriptorTypes 1}

atmNoClpNoScr  OBJECT-IDENTITY

```

```

STATUS      current
DESCRIPTION
  "This traffic descriptor is for no CLP and
  no Sustained Cell Rate.  The use of the
  parameter vector for this type:
  Parameter 1: CLP=0+1 peak cell rate in
               cells per second
  Parameter 2: not used
  Parameter 3: not used
  Parameter 4: not used
  Parameter 5: not used.
  This traffic descriptor type can be used
  for best effort traffic."
 ::= { atmTrafficDescriptorTypes 2}

```

```

atmClpNoTaggingNoScr OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION
    "This traffic descriptor is for no CLP without
    tagging and no Sustained Cell Rate.  The use
    of the parameter vector for this type:
    Parameter 1: CLP=0+1 peak cell rate in
                 cells per second
    Parameter 2: CLP=0 peak cell rate in
                 cells per second
    Parameter 3: not used
    Parameter 4: not used
    Parameter 5: not used."
    ::= { atmTrafficDescriptorTypes 3}

```

```

atmClpTaggingNoScr OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION
    "This traffic descriptor is for CLP with
    tagging and no Sustained Cell Rate.  The
    use of the parameter vector for this type:
    Parameter 1: CLP=0+1 peak cell rate in
                 cells per second
    Parameter 2: CLP=0 peak cell rate in
                 cells per second with excess
                 traffic tagged as CLP=1
    Parameter 3: not used
    Parameter 4: not used
    Parameter 5: not used."
    ::= { atmTrafficDescriptorTypes 4}

```

```

atmNoClpScr OBJECT-IDENTITY
  STATUS      current

```

## DESCRIPTION

"This traffic descriptor is for no CLP with Sustained Cell Rate. The use of the parameter vector for this type:  
 Parameter 1: CLP=0+1 peak cell rate in cells per second  
 Parameter 2: CLP=0+1 sustained cell rate in cells per second  
 Parameter 3: CLP=0+1 maximum burst size in cells  
 Parameter 4: not used  
 Parameter 5: not used."  
 ::= { atmTrafficDescriptorTypes 5 }

## atmClpNoTaggingScr OBJECT-IDENTITY

STATUS current

## DESCRIPTION

"This traffic descriptor is for CLP with Sustained Cell Rate and no tagging. The use of the parameter vector for this type:  
 Parameter 1: CLP=0+1 peak cell rate in cells per second  
 Parameter 2: CLP=0 sustained cell rate in cells per second  
 Parameter 3: CLP=0 maximum burst size in cells  
 Parameter 4: not used  
 Parameter 5: not used."  
 ::= { atmTrafficDescriptorTypes 6 }

## atmClpTaggingScr OBJECT-IDENTITY

STATUS current

## DESCRIPTION

"This traffic descriptor is for CLP with tagging and Sustained Cell Rate. The use of the parameter vector for this type:  
 Parameter 1: CLP=0+1 peak cell rate in cells per second  
 Parameter 2: CLP=0 sustained cell rate in cells per second with excess traffic tagged as CLP=1  
 Parameter 3: CLP=0 maximum burst size in cells  
 Parameter 4: not used  
 Parameter 5: not used."  
 ::= { atmTrafficDescriptorTypes 7 }

-- ATM Interface Configuration Parameters Group

```
-- This group contains ATM specific
-- configuration information associated with
-- an ATM interface beyond those
-- supported using the ifTable.
```

```
atmInterfaceConfTable    OBJECT-TYPE
    SYNTAX      SEQUENCE OF AtmInterfaceConfEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table contains ATM local interface
        configuration parameters, one entry per ATM
        interface port."
    ::= { atmMIBObjects 2 }
```

```
atmInterfaceConfEntry    OBJECT-TYPE
    SYNTAX      AtmInterfaceConfEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This list contains ATM interface configuration
        parameters and state variables."
    INDEX { ifIndex }
    ::= { atmInterfaceConfTable 1}
```

```
AtmInterfaceConfEntry    ::= SEQUENCE {
    atmInterfaceMaxVpcs      INTEGER,
    atmInterfaceMaxVccs      INTEGER,
    atmInterfaceConfVpcs     INTEGER,
    atmInterfaceConfVccs     INTEGER,
    atmInterfaceMaxActiveVpiBits  INTEGER,
    atmInterfaceMaxActiveVciBits  INTEGER,
    atmInterfaceIlmiVpi      INTEGER,
    atmInterfaceIlmiVci      INTEGER,
    atmInterfaceAddressType   INTEGER,
    atmInterfaceAdminAddress  OCTET STRING,
    atmInterfaceMyNeighborIpAddress  IpAddress,
    atmInterfaceMyNeighborIfName  DisplayString
}
```

```
atmInterfaceMaxVpcs      OBJECT-TYPE
    SYNTAX      INTEGER (0..4096)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The maximum number of VPCs (PVCs and SVCs)
        supported at this ATM interface.  At the ATM UNI,
```

the maximum number of VPCs (PVCs and SVCs)  
 ranges from 0 to 256 only."  
 ::= { atmInterfaceConfEntry 1}

atmInterfaceMaxVccs OBJECT-TYPE  
 SYNTAX INTEGER (0..65536)  
 MAX-ACCESS read-write  
 STATUS current  
 DESCRIPTION  
 "The maximum number of VCCs (PVCs and SVCs)  
 supported at this ATM interface."  
 ::= { atmInterfaceConfEntry 2}

atmInterfaceConfVpcs OBJECT-TYPE  
 SYNTAX INTEGER (0..4096)  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION  
 "The number of VPCs (PVCs and SVCs)  
 configured for use at this ATM interface.  
 At the ATM UNI, the configured number of  
 VPCs (PVCs and SVCs) can range from  
 0 to 256 only."  
 ::= { atmInterfaceConfEntry 3}

atmInterfaceConfVccs OBJECT-TYPE  
 SYNTAX INTEGER (0..65536)  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION  
 "The number of VCCs (PVCs and SVCs)  
 configured for use at the ATM interface."  
 ::= { atmInterfaceConfEntry 4}

atmInterfaceMaxActiveVpiBits OBJECT-TYPE  
 SYNTAX INTEGER (0..12)  
 MAX-ACCESS read-write  
 STATUS current  
 DESCRIPTION  
 "The maximum number of active VPI bits  
 configured for use at the ATM interface.  
 At the ATM UNI, the maximum number of active  
 VPI bits configured for use ranges from  
 0 to 8 only."  
 ::= { atmInterfaceConfEntry 5}

atmInterfaceMaxActiveVciBits OBJECT-TYPE  
 SYNTAX INTEGER (0..16)

```

MAX-ACCESS      read-write
STATUS          current
DESCRIPTION
  "The maximum number of active VCI bits
   configured for use at this ATM interface."
 ::= { atmInterfaceConfEntry 6}

```

```

atmInterfaceIlmiVpi      OBJECT-TYPE
SYNTAX                INTEGER (0..255)
MAX-ACCESS            read-write
STATUS                current
DESCRIPTION
  "The VPI value of the VCC supporting
   the ILMI at this ATM interface.  If the values of
   atmInterfaceIlmiVpi and atmInterfaceIlmiVci are
   both equal to zero then the ILMI is not
   supported at this ATM interface."
DEFVAL { 0 }
 ::= { atmInterfaceConfEntry 7}

```

```

atmInterfaceIlmiVci      OBJECT-TYPE
SYNTAX                INTEGER (0..65535)
MAX-ACCESS            read-write
STATUS                current
DESCRIPTION
  "The VCI value of the VCC supporting
   the ILMI at this ATM interface.  If the values of
   atmInterfaceIlmiVpi and atmInterfaceIlmiVci are
   both equal to zero then the ILMI is not
   supported at this ATM interface."
DEFVAL { 16 }
 ::= { atmInterfaceConfEntry 8}

```

```

atmInterfaceAddressType  OBJECT-TYPE
SYNTAX                INTEGER {
                        private(1),
                        nsapE164(2),
                        nativeE164(3),
                        other(4)
                        }
MAX-ACCESS            read-only
STATUS                current
DESCRIPTION
  "The type of primary ATM address configured
   for use at this ATM interface."
 ::= { atmInterfaceConfEntry 9 }

```

```

atmInterfaceAdminAddress OBJECT-TYPE

```



```

SYNTAX      OCTET STRING (SIZE(0..255))
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
  "An address assigned for administrative purposes,
  for example, an address associated with the
  service provider side of a public network UNI.
  If this interface has no assigned administrative
  address, or when the address used for
  administrative purposes is the same as that used
  for ifPhysAddress, then this is an octet string of
  zero length."
 ::= { atmInterfaceConfEntry 10 }

```

```

atmInterfaceMyNeighborIpAddress OBJECT-TYPE
SYNTAX      IPAddress
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
  "The IP address of the neighbor system connected to
  the far end of this interface, to which a Network
  Management Station can send SNMP messages, as IP
  datagrams sent to UDP port 161, in order to access
  network management information concerning the
  operation of that system. Note that the value
  of this object may be obtained in different ways,
  e.g., by manual configuration, or through ILMI
  interaction with the neighbor system."
 ::= { atmInterfaceConfEntry 11 }

```

```

atmInterfaceMyNeighborIfName      OBJECT-TYPE
SYNTAX      DisplayString
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
  "The textual name of the interface on the neighbor
  system on the far end of this interface, and to
  which this interface connects. If the neighbor
  system is manageable through SNMP and supports
  the object ifName, the value of this object must
  be identical with that of ifName for the ifEntry
  of the lowest level physical interface
  for this port. If this interface does not have a
  textual name, the value of this object is a zero
  length string. Note that the value of this object
  may be obtained in different ways, e.g., by manual
  configuration, or through ILMI interaction with
  the neighbor system."

```

```

 ::= { atmInterfaceConfEntry 12 }

-- The ATM Interface DS3 PLCP Group

-- This group contains the DS3 PLCP configuration and
-- state parameters of those ATM interfaces
-- which use DS3 PLCP for carrying ATM cells over DS3.

atmInterfaceDs3PlcpTable      OBJECT-TYPE
    SYNTAX          SEQUENCE OF AtmInterfaceDs3PlcpEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This table contains ATM interface DS3 PLCP
        parameters and state variables, one entry per
        ATM interface port."
    ::= { atmMIBObjects 3}

atmInterfaceDs3PlcpEntry      OBJECT-TYPE
    SYNTAX          AtmInterfaceDs3PlcpEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This list contains DS3 PLCP parameters and
        state variables at the ATM interface."
    INDEX          {ifIndex }
    ::= { atmInterfaceDs3PlcpTable 1}

AtmInterfaceDs3PlcpEntry      ::= SEQUENCE {
    atmInterfaceDs3PlcpSEFSs      Counter32,
    atmInterfaceDs3PlcpAlarmState  INTEGER,
    atmInterfaceDs3PlcpUASS      Counter32
    }

atmInterfaceDs3PlcpSEFSs      OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The number of DS3 PLCP Severely Errored Framing
        Second (SEFS). Each SEFS represents a
        one-second interval which contains
        one or more SEF event."
    ::= { atmInterfaceDs3PlcpEntry 1}

atmInterfaceDs3PlcpAlarmState  OBJECT-TYPE

```

```

SYNTAX          INTEGER {
                    noAlarm(1),
                    receivedFarEndAlarm(2),
                    incomingLOF(3)
                  }
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
  "This variable indicates if there is an
  alarm present for the DS3 PLCP.  The value
  receivedFarEndAlarm means that the DS3 PLCP
  has received an incoming Yellow
  Signal, the value incomingLOF means that
  the DS3 PLCP has declared a loss of frame (LOF)
  failure condition, and the value noAlarm
  means that there are no alarms present.
  Transition from the failure to the no alarm state
  occurs when no defects (e.g., LOF) are received
  for more than 10 seconds."
 ::= { atmInterfaceDs3PlcpEntry 2}

```

```

atmInterfaceDs3PlcpUASSs  OBJECT-TYPE
  SYNTAX          Counter32
  MAX-ACCESS      read-only
  STATUS          current
  DESCRIPTION
    "The counter associated with the number of
    Unavailable Seconds encountered by the PLCP."
  ::= { atmInterfaceDs3PlcpEntry 3}

```

-- The ATM Interface TC Sublayer Group

-- This group contains TC sublayer configuration and  
 -- state parameters of those ATM interfaces  
 -- which use TC sublayer for carrying ATM cells over  
 -- SONET or DS3.

```

atmInterfaceTCTable      OBJECT-TYPE
  SYNTAX          SEQUENCE OF AtmInterfaceTCEntry
  MAX-ACCESS      not-accessible
  STATUS          current
  DESCRIPTION
    "This table contains ATM interface TC
    Sublayer parameters and state variables,
    one entry per ATM interface port."
  ::= { atmMIBObjects 4}

```

```

atmInterfaceTCEntry      OBJECT-TYPE
    SYNTAX                AtmInterfaceTCEntry
    MAX-ACCESS            not-accessible
    STATUS                 current
    DESCRIPTION
        "This list contains TC Sublayer parameters
        and state variables at the ATM interface."
    INDEX {ifIndex }
    ::= { atmInterfaceTCTable 1}

AtmInterfaceTCEntry      ::= SEQUENCE {
    atmInterfaceOCDEvents Counter32,
    atmInterfaceTCAlarmState INTEGER
    }

atmInterfaceOCDEvents    OBJECT-TYPE
    SYNTAX                Counter32
    MAX-ACCESS            read-only
    STATUS                 current
    DESCRIPTION
        "The number of times the Out of Cell
        Delineation (OCD) events occur.  If seven
        consecutive ATM cells have Header Error
        Control (HEC) violations, an OCD event occurs.
        A high number of OCD events may indicate a
        problem with the TC Sublayer."
    ::= { atmInterfaceTCEntry 1}

atmInterfaceTCAlarmState OBJECT-TYPE
    SYNTAX                INTEGER {
                                noAlarm(1),
                                lcdFailure(2)
                                }
    MAX-ACCESS            read-only
    STATUS                 current
    DESCRIPTION
        "This variable indicates if there is an
        alarm present for the TC Sublayer.  The value
        lcdFailure indicates that a Loss of
        Cell Delineation (LCD) failure state has been
        declared for the TC Sublayer.  Transition from
        failure to the no alarm state occurs
        when 6 consecutive ATM cells
        are received with valid HEC, followed by
        about 10 seconds of acceptable working signal."
    ::= { atmInterfaceTCEntry 2}

```

```

-- ATM Traffic Descriptor Parameter Group

-- This group contains a set of self-consistent
-- ATM traffic parameters including the
-- ATM traffic QoS Class.

-- The ATM virtual link tables (i.e., VPL and VCL tables)
-- will use this ATM Traffic Descriptor table
-- to assign traffic parameters and QoS Class
-- to the receive and transmit directions of
-- the ATM virtual links (i.e., VPLs and VCLs).
-- The ATM VPL or VCL table will indicate a row
-- in the atmTrafficDescrParamTable
-- using its atmTrafficDescrParamIndex value.
-- The management application can then compare a set of
-- ATM traffic parameters with a single value.

-- If no suitable row(s) in the atmTrafficDescrParamTable
-- exists, the manager must create a new row(s) in this
-- table. If such a row is created, agent checks the
-- sanity of that set of ATM traffic parameter values.

-- When creating a new row, the parameter values
-- will be checked for self-consistency.
-- Predefined/template rows may be supported.

-- A row in the atmTrafficDescrParamTable is deleted
-- by setting the atmTrafficDescrRowStatus to destroy(6).
-- The agent will check whether this row is still in use
-- by any entry of the atmVplTable or atmVclTable.
-- The agent denies the request if the row is still in
-- use.

-- The ATM Traffic Descriptor Parameter Table

atmTrafficDescrParamTable    OBJECT-TYPE
    SYNTAX      SEQUENCE OF AtmTrafficDescrParamEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table contains information on ATM traffic
        descriptor type and the associated parameters."
    ::= { atmMIBObjects 5}

atmTrafficDescrParamEntry    OBJECT-TYPE
    SYNTAX      AtmTrafficDescrParamEntry
    MAX-ACCESS  not-accessible

```

```

STATUS          current
DESCRIPTION
  "This list contains ATM traffic descriptor
  type and the associated parameters."
INDEX {atmTrafficDescrParamIndex}
 ::= { atmTrafficDescrParamTable 1}

AtmTrafficDescrParamEntry ::= SEQUENCE {
  atmTrafficDescrParamIndex  AtmTrafficDescrParamIndex,
  atmTrafficDescrType        OBJECT IDENTIFIER,
  atmTrafficDescrParam1      Integer32,
  atmTrafficDescrParam2      Integer32,
  atmTrafficDescrParam3      Integer32,
  atmTrafficDescrParam4      Integer32,
  atmTrafficDescrParam5      Integer32,
  atmTrafficQoSClass         INTEGER,
  atmTrafficDescrRowStatus   RowStatus
}

atmTrafficDescrParamIndex  OBJECT-TYPE
SYNTAX                     AtmTrafficDescrParamIndex
MAX-ACCESS                 not-accessible
STATUS                     current
DESCRIPTION
  "This object is used by the virtual link
  table (i.e., VPL or VCL table)
  to identify the row of this table."
 ::= { atmTrafficDescrParamEntry 1}

atmTrafficDescrType        OBJECT-TYPE
SYNTAX                     OBJECT IDENTIFIER
MAX-ACCESS                 read-create
STATUS                     current
DESCRIPTION
  "The value of this object identifies the type
  of ATM traffic descriptor.
  The type may indicate no traffic descriptor or
  traffic descriptor with one or more parameters.
  These parameters are specified as a parameter
  vector, in the corresponding instances of the
  objects:
    atmTrafficDescrParam1
    atmTrafficDescrParam2
    atmTrafficDescrParam3
    atmTrafficDescrParam4
    atmTrafficDescrParam5."
DEFVAL { atmNoTrafficDescriptor }
 ::= { atmTrafficDescrParamEntry 2}

```

```

atmTrafficDescrParam1    OBJECT-TYPE
    SYNTAX                Integer32
    MAX-ACCESS            read-create
    STATUS                 current
    DESCRIPTION
        "The first parameter of the ATM traffic descriptor
        used according to the value of
        atmTrafficDescrType."
    DEFVAL { 0 }
    ::= { atmTrafficDescrParamEntry 3}

atmTrafficDescrParam2    OBJECT-TYPE
    SYNTAX                Integer32
    MAX-ACCESS            read-create
    STATUS                 current
    DESCRIPTION
        "The second parameter of the ATM traffic descriptor
        used according to the value of
        atmTrafficDescrType."
    DEFVAL { 0 }
    ::= { atmTrafficDescrParamEntry 4}

atmTrafficDescrParam3    OBJECT-TYPE
    SYNTAX                Integer32
    MAX-ACCESS            read-create
    STATUS                 current
    DESCRIPTION
        "The third parameter of the ATM traffic descriptor
        used according to the value of
        atmTrafficDescrType."
    DEFVAL { 0 }
    ::= { atmTrafficDescrParamEntry 5}

atmTrafficDescrParam4    OBJECT-TYPE
    SYNTAX                Integer32
    MAX-ACCESS            read-create
    STATUS                 current
    DESCRIPTION
        "The fourth parameter of the ATM traffic descriptor
        used according to the value of
        atmTrafficDescrType."
    DEFVAL { 0 }
    ::= { atmTrafficDescrParamEntry 6}

atmTrafficDescrParam5    OBJECT-TYPE
    SYNTAX                Integer32
    MAX-ACCESS            read-create
    STATUS                 current

```

## DESCRIPTION

"The fifth parameter of the ATM traffic descriptor used according to the value of atmTrafficDescrType."

DEFVAL { 0 }

::= { atmTrafficDescrParamEntry 7 }

atmTrafficQoSClass OBJECT-TYPE

SYNTAX INTEGER (0..255)

MAX-ACCESS read-create

STATUS current

## DESCRIPTION

"The value of this object identifies the QoS Class. Four Service classes have been specified in the ATM Forum UNI Specification: Service Class A: Constant bit rate video and Circuit emulation Service Class B: Variable bit rate video/audio Service Class C: Connection-oriented data Service Class D: Connectionless data Four QoS classes numbered 1, 2, 3, and 4 have been specified with the aim to support service classes A, B, C, and D respectively. An unspecified QoS Class numbered '0' is used for best effort traffic."

DEFVAL { 0 }

::= { atmTrafficDescrParamEntry 8 }

atmTrafficDescrRowStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

## DESCRIPTION

"This object is used to create a new row or modify or delete an existing row in this table."

DEFVAL { active }

::= { atmTrafficDescrParamEntry 9 }

-- ATM Interface Virtual Path Link (VPL) Group

-- This group contains configuration and state  
 -- information of a bi-directional Virtual Path Link  
 -- (VPL)

-- This table can be used to create, delete or modify



```
-- a VPL that is terminated in an ATM host or switch.
-- This table can also be used to create, delete or
-- modify a VPL which is cross-connected to another
-- VPL.

-- In the example below, the traffic flows on the receive
-- and transmit directions of the VPLs are characterized
-- by atmVplReceiveTrafficDescrIndex and
-- atmVplTransmitTrafficDescrIndex respectively.
-- The cross-connected VPLs are identified by
-- atmVplCrossConnectIdentifier.
```

```
--
--
-- VPL |-----| VPL
-- receive | ATM Host, Switch, or Network | receive
-- =====> X | | X <=====
-- <===== X | | X =====>
-- transmit | | | transmit
--
```

```
-- The ATM Interface VPL Table
```

```
atmVplTable OBJECT-TYPE
    SYNTAX SEQUENCE OF AtmVplEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The Virtual Path Link (VPL) table. A
        bi-directional VPL is modeled as one entry
        in this table."
    ::= { atmMIBObjects 6}
```

```
atmVplEntry OBJECT-TYPE
    SYNTAX AtmVplEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the VPL table. This entry is
        used to model a bi-directional VPL.
        To create a VPL at an ATM interface,
        either of the following procedures are used:
```

## Negotiated VPL establishment

- (1) The management application creates a VPL entry in the atmVplTable by setting atmVplRowStatus to createAndWait(5). This may fail for the following reasons:
  - The selected VPI value is unavailable,
  - The selected VPI value is in use.Otherwise, the agent creates a row and reserves the VPI value on that port.
- (2) The manager selects an existing row(s) in the atmTrafficDescrParamTable, thereby, selecting a set of self-consistent ATM traffic parameters and the QoS Class for receive and transmit directions of the VPL.
  - (2a) If no suitable row(s) in the atmTrafficDescrParamTable exists, the manager must create a new row(s) in that table.
  - (2b) The manager characterizes the VPL's traffic parameters through setting the atmVplReceiveTrafficDescrIndex and the atmVplTransmitTrafficDescrIndex values in the VPL table, which point to the rows containing desired ATM traffic parameter values in the atmTrafficDescrParamTable. The agent will check the availability of resources and may refuse the request.
- (3) The manager activates the VPL by setting the atmVplRowStatus to active(1). If this set is successful, the agent has reserved the resources to satisfy the requested traffic parameter values and the QoS Class for that VPL.
- (4) If the VPL terminates a VPC in the ATM host or switch, the manager turns on the atmVplAdminStatus to up(1) to turn the VPL traffic flow on. Otherwise, the atmVpCrossConnectTable must be used to cross-connect the VPL to another VPL(s) in an ATM switch or network.

## One-Shot VPL Establishment

A VPL may also be established in one step by a set-request with all necessary VPL parameter values and atmVplRowStatus set to createAndGo(4).

In contrast to the negotiated VPL establishment which allows for detailed error checking (i.e., set errors are explicitly linked to particular resource acquisition failures), the one-shot VPL establishment performs the setup on one operation but does not have the advantage of step-wise error checking.

#### VPL Retirement

A VPL is released by setting atmVplRowStatus to destroy(6), and the agent may release all associated resources."

```
INDEX {ifIndex, atmVplVpi }
 ::= { atmVplTable 1}
```

```
AtmVplEntry ::= SEQUENCE {
    atmVplVpi                INTEGER,
    atmVplAdminStatus        INTEGER,
    atmVplOperStatus         INTEGER,
    atmVplLastChange         TimeStamp,
    atmVplReceiveTrafficDescrIndex
                            AtmTrafficDescrParamIndex,
    atmVplTransmitTrafficDescrIndex
                            AtmTrafficDescrParamIndex,
    atmVplCrossConnectIdentifier INTEGER,
    atmVplRowStatus           RowStatus
}
```

```
atmVplVpi OBJECT-TYPE
    SYNTAX      INTEGER (1..4095)
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "The VPI value of the VPL.
         Note that the VPI=0 is not used
         for a VPL not associated with a VCL.
         The maximum VPI value cannot
         exceed the value allowable by the
         atmInterfaceMaxVpiBits."
    ::= { atmVplEntry 1}
```

```

atmVplAdminStatus    OBJECT-TYPE
    SYNTAX    INTEGER {
                    up(1),
                    down(2)
                }
    MAX-ACCESS    read-create
    STATUS        current
    DESCRIPTION
        "This object is implemented only for a VPL which
        terminates a VPC (i.e.,
        one which is NOT cross-connected to other VPLs).
        Its value specifies the desired administrative
        state of the VPL. The up and down states indicate
        that the traffic flow is enabled and disabled
        respectively for this VPL."
    DEFVAL { down }
    ::= { atmVplEntry 2}

```

```

atmVplOperStatus    OBJECT-TYPE
    SYNTAX    INTEGER {
                    up(1),
                    down(2),
                    unknown(3)
                }
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "This object indicates the current operational
        status of the VPL. The up and down states
        indicate that the VPL is currently
        operational, or not operational, respectively.
        The unknown state indicates that the status of
        this VPL cannot be determined."
    ::= { atmVplEntry 3}

```

```

atmVplLastChange    OBJECT-TYPE
    SYNTAX    TimeStamp
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "The value of MIB II's sysUpTime object
        at the time this VPL entered its current
        operational state. If the current state was
        entered prior to the last re-initialization of the
        agent, then this object contains a zero value."
    ::= { atmVplEntry 4 }

```

```

atmVplReceiveTrafficDescrIndex    OBJECT-TYPE
    SYNTAX          AtmTrafficDescrParamIndex
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        "The value of this object identifies the row
         in the atmTrafficDescrParamTable which
         applies to the receive direction of the VPL."
        ::= { atmVplEntry 5}

atmVplTransmitTrafficDescrIndex    OBJECT-TYPE
    SYNTAX          AtmTrafficDescrParamIndex
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        "The value of this object identifies the row
         in the atmTrafficDescrParamTable which
         applies to the transmit direction of the VPL."
        ::= { atmVplEntry 6}

atmVplCrossConnectIdentifier      OBJECT-TYPE
    SYNTAX          INTEGER (0..2147483647)
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object is implemented only for a VPL
         which is cross-connected to other VPLs
         that belong to the same VPC. All such
         associated VPLs have the same value of this
         object, and all their cross-connections are
         identified by entries in the
         atmVpCrossConnectTable for which
         atmVpCrossConnectIndex has the same value.
         The value of this object is initialized by the
         agent after the associated entries in the
         atmVpCrossConnectTable have been created."
        ::= {atmVplEntry 7}

atmVplRowStatus                  OBJECT-TYPE
    SYNTAX          RowStatus
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        "This object is used to create, delete
         or modify a row in this table.
         To create a new VCL, this object is
         initially set to 'createAndWait' or
         'createAndGo'. This object must not be

```

```

    set to 'active' unless the following columnar
    objects exist in this row:
    atmVplReceiveTrafficDescrIndex and
    atmVplTransmitTrafficDescrIndex."
    DEFVAL { active }
    ::= { atmVplEntry 8}

```

```
-- ATM Interface Virtual Channel Link (VCL) Group
```

```
-- This group contains configuration and state
-- information of a bi-directional Virtual Channel
-- Link (VCL) at an ATM interface.
```

```
-- This table can be used to create, delete or modify
-- a VCL that is terminated in an ATM host or switch.
-- This table can also be
-- used to create, delete or modify a VCL that is
-- cross-connected to another VCL.
```

```
-- The ATM Interface VCL Table
```

```

atmVclTable      OBJECT-TYPE
    SYNTAX          SEQUENCE OF AtmVclEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "The Virtual Channel Link (VCL) table. A
        bi-directional VCL is modeled as one entry
        in this table."
    ::= { atmMIBObjects 7}

```

```

atmVclEntry      OBJECT-TYPE
    SYNTAX          AtmVclEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "An entry in the VCL table. This entry is
        used to model a bi-directional VCL.
        To create a VCL at an ATM interface,
        either of the following procedures are used:

```

```

    Negotiated VCL establishment

```

```

    (1) The management application creates
        a VCL entry in the atmVclTable

```

by setting atmVclRowStatus to createAndWait(5). This may fail for the following reasons:

- The selected VPI/VCI values are unavailable,
- The selected VPI/VCI values are in use.

Otherwise, the agent creates a row and reserves the VPI/VCI values on that port.

- (2) The manager selects an existing row(s) in the atmTrafficDescrParamTable, thereby, selecting a set of self-consistent ATM traffic parameters and the QoS Class for receive and transmit directions of the VCL.
- (2a) If no suitable row(s) in the atmTrafficDescrParamTable exists, the manager must create a new row(s) in that table.
- (2b) The manager characterizes the VCL's traffic parameters through setting the atmVclReceiveTrafficDescrIndex and the atmVclTransmitTrafficDescrIndex values in the VCL table, which point to the rows containing desired ATM traffic parameter values in the atmTrafficDescrParamTable. The agent will check the availability of resources and may refuse the request.
- (3) The manager activates the VCL by setting the atmVclRowStatus to active(1). If this set is successful, the agent has reserved the resources to satisfy the requested traffic parameter values and the QoS Class for that VCL.
- (4) If the VCL terminates a VCC in the ATM host or switch, the manager turns on the atmVclAdminStatus to up(1) to turn the VCL traffic flow on. Otherwise, the atmVcCrossConnectTable must be used to cross-connect the VCL to another VCL(s) in an ATM switch or network.

#### One-Shot VCL Establishment

A VCL may also be established in one step by a set-request with all necessary VCL parameter values and atmVclRowStatus set to createAndGo(4).

In contrast to the negotiated VCL establishment which allows for detailed error checking (i.e., set errors are explicitly linked to particular resource acquisition failures), the one-shot VCL establishment performs the setup on one operation but does not have the advantage of step-wise error checking.

#### VCL Retirement

A VCL is released by setting atmVclRowStatus to destroy(6), and the agent may release all associated resources."

```
INDEX {ifIndex, atmVclVpi, atmVclVci }
 ::= { atmVclTable 1}
```

```
AtmVclEntry ::= SEQUENCE {
    atmVclVpi          INTEGER,
    atmVclVci         INTEGER,
    atmVclAdminStatus INTEGER,
    atmVclOperStatus  INTEGER,
    atmVclLastChange  TimeStamp,
    atmVclReceiveTrafficDescrIndex
                        AtmTrafficDescrParamIndex,
    atmVclTransmitTrafficDescrIndex
                        AtmTrafficDescrParamIndex,
    atmVccAalType     INTEGER,
    atmVccAal5CpcsTransmitSduSize  INTEGER,
    atmVccAal5CpcsReceiveSduSize   INTEGER,
    atmVccAal5EncapsType            INTEGER,
    atmVclCrossConnectIdentifier    INTEGER,
    atmVclRowStatus                 RowStatus
}
```

```
atmVclVpi OBJECT-TYPE
    SYNTAX      INTEGER (0..4095)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The VPI value of the VCL.
         The maximum VPI value cannot
         exceed the value allowable by the
         atmInterfaceMaxVpiBits."
    ::= { atmVclEntry 1}
```

```
atmVclVci OBJECT-TYPE
    SYNTAX      INTEGER (0..65535)
```



```

MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
  "The VCI value of the VCL.
  The maximum VCI value cannot
  exceed the value allowable by the
  atmInterfaceMaxVciBits."
 ::= { atmVclEntry 2}

```

```

atmVclAdminStatus  OBJECT-TYPE
SYNTAX  INTEGER  {
                    up(1),
                    down(2)
                }
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION
  "This object is implemented only for a VCL which
  terminates a VCC (i.e.,
  one which is NOT cross-connected to other VCLs).
  Its value specifies the desired administrative
  state of the VCL.  The up and down states indicate
  that the traffic flow is enabled and disabled
  respectively for this VCL."
 ::= { atmVclEntry 3}

```

```

atmVclOperStatus  OBJECT-TYPE
SYNTAX  INTEGER  {
                    up(1),
                    down(2),
                    unknown(3)
                }
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
  "This object indicates the current operational
  status of the VCL.  The up and down states
  indicate that the VCL is currently
  operational, or not operational, respectively.
  The unknown state indicates that the status of
  this VCL cannot be determined."
 ::= { atmVclEntry 4}

```

```

atmVclLastChange  OBJECT-TYPE
SYNTAX  TimeStamp
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION

```

"The value of MIB II's sysUpTime object at the time this VCL entered its current operational state. If the current state was entered prior to the last re-initialization of the agent, then this object contains a zero value."  
 ::= { atmVclEntry 5 }

atmVclReceiveTrafficDescrIndex      OBJECT-TYPE  
 SYNTAX                              AtmTrafficDescrParamIndex  
 MAX-ACCESS                          read-create  
 STATUS                                current  
 DESCRIPTION  
 "The value of this object identifies the row in the ATM Traffic Descriptor Table which applies to the receive direction of this VCL."  
 ::= { atmVclEntry 6 }

atmVclTransmitTrafficDescrIndex      OBJECT-TYPE  
 SYNTAX                              AtmTrafficDescrParamIndex  
 MAX-ACCESS                          read-create  
 STATUS                                current  
 DESCRIPTION  
 "The value of this object identifies the row of the ATM Traffic Descriptor Table which applies to the transmit direction of this VCL."  
 ::= { atmVclEntry 7 }

atmVccAalType                          OBJECT-TYPE  
 SYNTAX                              INTEGER {  
                                     aal1(1),  
                                     aal34(2),  
                                     aal5(3),  
                                     other(4),  
                                     unknown(5)  
                                     }  
 MAX-ACCESS                          read-create  
 STATUS                                current  
 DESCRIPTION  
 "An instance of this object only exists when the local VCL end-point is also the VCC end-point, and AAL is in use.  
 The type of AAL used on this VCC.  
 The AAL type includes AAL1, AAL3/4, and AAL5. The other(4) may be user-defined AAL type. The unknown type indicates that the AAL type cannot be determined."  
 ::= { atmVclEntry 8 }

```

atmVccAal5CpcsTransmitsSduSize OBJECT-TYPE
    SYNTAX      INTEGER (1..65535)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "An instance of this object only exists when the
        local VCL end-point is also the VCC end-point,
        and AAL5 is in use.
        The maximum AAL5 CPCS SDU size in octets that is
        supported on the transmit direction of this VCC."
    DEFVAL { 9188 }
        ::= { atmVclEntry 9 }

atmVccAal5CpcsReceiveSduSize OBJECT-TYPE
    SYNTAX      INTEGER (1..65535)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "An instance of this object only exists when the
        local VCL end-point is also the VCC end-point,
        and AAL5 is in use.
        The maximum AAL5 CPCS SDU size in octets that is
        supported on the receive direction of this VCC."
    DEFVAL { 9188 }
        ::= { atmVclEntry 10 }

atmVccAal5EncapsType OBJECT-TYPE
    SYNTAX      INTEGER {
        vcMultiplexRoutedProtocol(1),
        vcMultiplexBridgedProtocol8023(2),
        vcMultiplexBridgedProtocol8025(3),
        vcMultiplexBridgedProtocol8026(4),
        vcMultiplexLANemulation8023(5),
        vcMultiplexLANemulation8025(6),
        llcEncapsulation(7),
        multiprotocolFrameRelaySscs(8),
        other(9),
        unknown(10)
        }
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "An instance of this object only exists when the
        local VCL end-point is also the VCC end-point,
        and AAL5 is in use.
        The type of data encapsulation used over
        the AAL5 SCS layer. The definitions reference
        RFC 1483 Multiprotocol Encapsulation

```

over ATM AAL5 and to the ATM Forum  
LAN Emulation specification."  
DEFVAL { llcEncapsulation }  
      ::= { atmVclEntry 11 }

atmVclCrossConnectIdentifier   OBJECT-TYPE  
SYNTAX                   INTEGER (0..2147483647)  
MAX-ACCESS               read-only  
STATUS                    current  
DESCRIPTION  
  "This object is implemented only for a VCL  
  which is cross-connected to other VCLs  
  that belong to the same VCC. All such  
  associated VCLs have the same value of this  
  object, and all their cross-connections are  
  identified by entries in the  
  atmVcCrossConnectTable for which  
  atmVcCrossConnectIndex has the same value.  
  The value of this object is initialized by the  
  agent after the associated entries in the  
  atmVcCrossConnectTable have been created."  
  ::= {atmVclEntry 12}

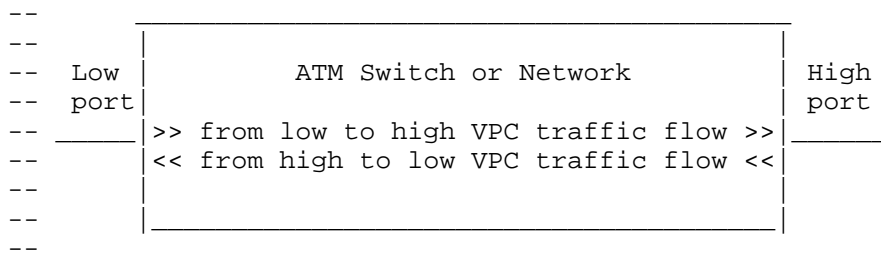
atmVclRowStatus                OBJECT-TYPE  
SYNTAX                    RowStatus  
MAX-ACCESS                read-create  
STATUS                    current  
DESCRIPTION  
  "This object is used to create, delete or  
  modify a row in this table. To create  
  a new VCL, this object is initially set  
  to 'createAndWait' or 'createAndGo'.  
  This object must not be set to 'active'  
  unless the following columnar objects exist  
  in this row:  
  atmVclReceiveTrafficDescrIndex,  
  atmVclTransmitTrafficDescrIndex.  
  In addition, if the local VCL end-point  
  is also the VCC end-point:  
  atmVccAalType.  
  In addition, for AAL5 connections only:  
  atmVccAal5CpcsTransmitSduSize,  
  atmVccAal5CpcsReceiveSduSize, and  
  atmVccAal5EncapsType."  
  DEFVAL { active }  
  ::= {atmVclEntry 13}

```
--      ATM Virtual Path (VP) Cross Connect Group

-- This group contains configuration and state
-- information of all point-to-point,
-- point-to-multipoint, or multipoint-to-multipoint
-- VP cross-connects.

-- This table has read-create access and can be used
-- to cross-connect the VPLs together in an ATM switch
-- or network. The atmVpCrossConnectIndex
-- is used to associate the related
-- VPLs that are cross-connected together.

-- The ATM VP Cross Connect Group
-- models each bi-directional VPC
-- cross-connect as a set of entries in
-- the atmVpCrossConnectTable. A
-- point-to-point VPC cross-connect is modeled
-- as one entry; a point-to-multipoint (N leafs) VPC
-- cross-connect as N entries in this table; and
-- a multipoint-to-multipoint (N parties) VPC cross-
-- connect as N(N-1)/2 entries in this table.
-- In the latter cases, all the N (or N(N-1)/2) entries
-- are associated with a single VPC cross-connect by
-- having the same value of atmVpCrossConnectIndex.
```



```
-- The terms low and high are chosen to represent
-- numerical ordering of the two interfaces associated
-- with a VPC cross-connect. That is, the ATM interface
-- with the lower value of ifIndex is termed 'low',
-- while the other ATM interface associated with the
-- VPC cross-connect is termed 'high'. This terminology
-- is used to provide directional information; for
-- example, the atmVpCrossConnectL2HOperStatus applies
-- to the low->high direction, and
-- atmVpCrossConnectH2LOperStatus applies to the
-- high->low direction, as illustrated above.
```

```

atmVpCrossConnectIndexNext    OBJECT-TYPE
    SYNTAX  INTEGER (0..2147483647)
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "This object contains an appropriate value to
        be used for atmVpCrossConnectIndex when creating
        entries in the atmVpCrossConnectTable.  The value
        0 indicates that no unassigned entries are
        available.  To obtain the atmVpCrossConnectIndex
        value for a new entry, the manager issues a
        management protocol retrieval operation to obtain
        the current value of this object.  After each
        retrieval, the agent should modify the value to
        the next unassigned index."
    ::= { atmMIBObjects 8 }

```

-- The ATM VP Cross Connect Table

```

atmVpCrossConnectTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF AtmVpCrossConnectEntry
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
        "The ATM VP Cross Connect table.  A bi-
        directional VP cross-connect which
        cross-connects two VPLs is modeled
        as one entry in this table."
    ::= { atmMIBObjects 9 }

```

```

atmVpCrossConnectEntry OBJECT-TYPE
    SYNTAX  AtmVpCrossConnectEntry
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
        "An entry in the ATM VP Cross Connect table.
        This entry is used to model a bi-directional
        ATM VP cross-connect which cross-connects
        two VPLs."

```

Step-wise Procedures to set up a VP Cross-connect

Once the entries in the atmVplTable are created,  
the following procedures are used  
to cross-connect the VPLs together.

- (1) The manager obtains a unique `atmVpCrossConnectIndex` by reading the `atmVpCrossConnectIndexNext` object.
- (2) Next, the manager creates a set of one or more rows in the ATM VP Cross Connect Table, one for each cross-connection between two VPLs. Each row is indexed by the ATM interface port numbers and VPI values of the two ends of that cross-connection. This set of rows specifies the topology of the VPC cross-connect and is identified by a single value of `atmVpCrossConnectIndex`.

#### Negotiated VP Cross-Connect Establishment

- (2a) The manager creates a row in this table by setting `atmVpCrossConnectRowStatus` to `createAndWait(5)`. The agent checks the requested topology and the mutual sanity of the ATM traffic parameters and QoS Classes, i.e., the row creation fails if:
  - the requested topology is not supported by the agent,
  - the traffic/QoS parameter values associated with the requested row are incompatible with those of already existing rows for this VP cross-connect.[For example, for setting up a point-to-point VP cross-connect, the ATM traffic parameters in the receive direction of a VPL at the low end of the cross-connect must equal to the traffic parameters in the transmit direction of the other VPL at the high end of the cross-connect, otherwise, the row creation fails.] The agent also checks for internal errors in building the cross-connect.

The `atmVpCrossConnectIndex` values in the corresponding `atmVplTable` rows are filled in by the agent at this point.

- (2b) The manager promotes the row in the `atmVpCrossConnectTable` by setting `atmVpCrossConnectRowStatus` to `active(1)`. If this set is successful, the agent has reserved the resources specified by the ATM traffic

parameter and QoS Class values for each direction of the VP cross-connect in an ATM switch or network.

- (3) The manager sets the `atmVpCrossConnectAdminStatus` to `up(1)` in all rows of this VP cross-connect to turn the traffic flow on.

#### One-Shot VP Cross-Connect Establishment

A VP cross-connect may also be established in one step by a `set-request` with all necessary parameter values and `atmVpCrossConnectRowStatus` set to `createAndGo(4)`.

In contrast to the negotiated VP cross-connect establishment which allows for detailed error checking (i.e., set errors are explicitly linked to particular resource acquisition failures), the one-shot VP cross-connect establishment performs the setup on one operation but does not have the advantage of step-wise error checking.

#### VP Cross-Connect Retirement

A VP cross-connect identified by a particular value of `atmVpCrossConnectIndex` is released by:

- (1) Setting `atmVpCrossConnectRowStatus` of all rows identified by this value of `atmVpCrossConnectIndex` to `destroy(6)`. The agent may release all associated resources, and the `atmVpCrossConnectIndex` values in the corresponding `atmVplTable` row are removed. Note that a situation when only a subset of the associated rows are deleted corresponds to a VP topology change.
- (2) After deletion of the appropriate `atmVpCrossConnectEntries`, the manager may set `atmVplRowStatus` to `destroy(6)` the associated VPLs. The agent releases the resources and removes the associated rows in the `atmVplTable`.



## VP Cross-connect Reconfiguration

At the discretion of the agent, a VP cross-connect may be reconfigured by adding and/or deleting leafs to/from the VP topology as per the VP cross-connect establishment/retirement procedures. Reconfiguration of traffic/QoS parameter values requires release of the VP cross-connect before those parameter values may be changed for individual VPLs."

```
INDEX { atmVpCrossConnectIndex,
        atmVpCrossConnectLowIfIndex,
        atmVpCrossConnectLowVpi,
        atmVpCrossConnectHighIfIndex,
        atmVpCrossConnectHighVpi }
 ::= { atmVpCrossConnectTable 1 }
```

```
AtmVpCrossConnectEntry ::= SEQUENCE {
    atmVpCrossConnectIndex      INTEGER,
    atmVpCrossConnectLowIfIndex IfIndex,
    atmVpCrossConnectLowVpi     INTEGER,
    atmVpCrossConnectHighIfIndex IfIndex,
    atmVpCrossConnectHighVpi    INTEGER,
    atmVpCrossConnectAdminStatus INTEGER,
    atmVpCrossConnectL2HOperStatus INTEGER,
    atmVpCrossConnectH2LOperStatus INTEGER,
    atmVpCrossConnectL2HLastChange TimeStamp,
    atmVpCrossConnectH2LLastChange TimeStamp,
    atmVpCrossConnectRowStatus   RowStatus
}
```

```
atmVpCrossConnectIndex OBJECT-TYPE
    SYNTAX  INTEGER (1..2147483647)
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
        "A unique value to identify this VP cross-connect."
    ::= { atmVpCrossConnectEntry 1 }
```

```
atmVpCrossConnectLowIfIndex OBJECT-TYPE
    SYNTAX  IfIndex
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
        "The value of this object is equal to MIB II's
         ifIndex value of the ATM interface port for this
         VP cross-connect. The term low implies
```

that this ATM interface has the numerically lower ifIndex value than the other ATM interface identified in the same atmVpCrossConnectEntry."  
 ::= { atmVpCrossConnectEntry 2 }

atmVpCrossConnectLowVpi OBJECT-TYPE

SYNTAX INTEGER (1..4095)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The value of this object is equal to the VPI value at the ATM interface associated with the VP cross-connect that is identified by atmVpCrossConnectLowIfIndex. The VPI value cannot exceed the number supported by the atmInterfaceMaxVpiBits at the low ATM interface port."

::= { atmVpCrossConnectEntry 3 }

atmVpCrossConnectHighIfIndex OBJECT-TYPE

SYNTAX IfIndex

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The value of this object is equal to MIB II's ifIndex value of the ATM interface port for this VP cross-connect. The term high implies that this ATM interface has the numerically higher ifIndex value than the other ATM interface identified in the same atmVpCrossConnectEntry."

::= { atmVpCrossConnectEntry 4 }

atmVpCrossConnectHighVpi OBJECT-TYPE

SYNTAX INTEGER (1..4095)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The value of this object is equal to the VPI value at the ATM interface associated with the VP cross-connect that is identified by atmVpCrossConnectHighIfIndex. The VPI value cannot exceed the number supported by the atmInterfaceMaxVpiBits at the high ATM interface port."

::= { atmVpCrossConnectEntry 5 }

atmVpCrossConnectAdminStatus OBJECT-TYPE

```

SYNTAX  INTEGER {
                up(1),
                down(2)
            }
MAX-ACCESS  read-create
STATUS  current
DESCRIPTION
    "The value of this object identifies the desired
    administrative status of this bi-directional
    VP cross-connect.  The up and down states
    indicate that the traffic flow is enabled
    and disabled respectively on this VP
    cross-connect."
DEFVAL { down }
 ::= { atmVpCrossConnectEntry 6 }

```

atmVpCrossConnectL2HOperStatus OBJECT-TYPE

```

SYNTAX  INTEGER {
                up(1),
                down(2),
                unknown(3)
            }
MAX-ACCESS  read-only
STATUS  current
DESCRIPTION
    "The value of this object identifies the current
    operational status of the VP cross-connect
    in one direction; (i.e., from the low to
    high direction).  The up and down states indicate
    that this ATM VP cross-connect from low
    to high direction is operational or not
    operational respectively.
    The unknown state indicates that
    the state of it cannot be determined."
 ::= { atmVpCrossConnectEntry 7 }

```

atmVpCrossConnectH2LOperStatus OBJECT-TYPE

```

SYNTAX  INTEGER {
                up(1),
                down(2),
                unknown(3)
            }
MAX-ACCESS  read-only
STATUS  current
DESCRIPTION
    "The value of this object identifies the current
    operational status of the VP cross-connect
    in one direction; (i.e., from the high to

```

low direction). The up and down states indicate that this ATM VP cross-connect from high to low direction is operational or not operational respectively. The unknown state indicates that the state of it cannot be determined."  
 ::= { atmVpCrossConnectEntry 8 }

atmVpCrossConnectL2HLastChange OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of MIB II's sysUpTime object at the time this VP cross-connect entered its current operational state in the low to high direction. If the current state was entered prior to the last re-initialization of the agent, then this object contains a zero value."

::= { atmVpCrossConnectEntry 9 }

atmVpCrossConnectH2LLastChange OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of MIB II's sysUpTime object at the time this VP cross-connect entered its current operational state in the high to low direction. If the current state was entered prior to the last re-initialization of the agent, then this object contains a zero value."

::= { atmVpCrossConnectEntry 10 }

atmVpCrossConnectRowStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The status of this entry in the atmVpCrossConnectTable. This object is used to create a cross-connect for cross-connecting VPLs which are created using the atmVplTable or to change or delete an existing cross-connect. This object must be initially set to 'createAndWait' or 'createAndGo'. This object cannot be set to 'active' unless the following columnar object exists

```

    in this row: atmVpCrossConnectAdminStatus.
    To turn on a VP cross-connect,
    the atmVpCrossConnectAdminStatus
    is set to 'up'."
DEFVAL { active }
 ::= { atmVpCrossConnectEntry 11 }

--      ATM Virtual Channel (VC) Cross Connect Group

-- This group contains configuration and state
-- information of a bi-directional VC cross-connect.

-- This group is used to model a bi-directional
-- point-to-point, point-to-multipoint or
-- multipoint-to-multipoint VC cross-connects.

-- This table has read-create access and is used
-- to cross-connect the VCLs together in an ATM switch
-- or network that belong to a VC connection.
-- The atmVcCrossConnectIndex is used to associate
-- the related VCLs that are cross-connected together.

-- The step-wise procedures described for setting
-- up a VP cross-connect are also used for setting up
-- a VC cross-connect.

atmVcCrossConnectIndexNext      OBJECT-TYPE
SYNTAX  INTEGER (0..2147483647)
MAX-ACCESS  read-only
STATUS    current
DESCRIPTION
    "This object contains an appropriate value to
    be used for atmVcCrossConnectIndex when creating
    entries in the atmVcCrossConnectTable. The value
    0 indicates that no unassigned entries are
    available. To obtain the atmVpCrossConnectIndex
    value for a new entry, the manager issues a
    management protocol retrieval operation to obtain
    the current value of this object. After each
    retrieval, the agent should modify the value to
    the next unassigned index."
 ::= { atmMIBObjects 10 }

-- The ATM VC Cross Connect Table

```

```

atmVcCrossConnectTable OBJECT-TYPE
    SYNTAX SEQUENCE OF AtmVcCrossConnectEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The ATM VC Cross Connect table.  A bi-
        directional VC cross-connect which
        cross-connects two end points (i.e., VCLs)
        is modeled as one entry in this table."
    ::= { atmMIBObjects 11 }

```

atmVcCrossConnectEntry OBJECT-TYPE

```

SYNTAX AtmVcCrossConnectEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    "An entry in the ATM VC Cross Connect table.
    This entry is used to model a bi-directional ATM
    VC cross-connect cross-connecting two end points.

    Step-wise Procedures to set up a VC Cross-connect

    Once the entries in the atmVclTable are created,
    the following procedures are used
    to cross-connect the VCLs together to
    form a VCC segment.

    (1) The manager obtains a unique
        atmVcCrossConnectIndex by reading the
        atmVcCrossConnectIndexNext object.

    (2) Next, the manager creates a set of one
        or more rows in the ATM VC Cross Connect
        Table, one for each cross-connection between
        two VCLs.  Each row is indexed by the ATM
        interface port numbers and VPI/VCI values of
        the two ends of that cross-connection.
        This set of rows specifies the topology of the
        VCC cross-connect and is identified by a single
        value of atmVcCrossConnectIndex.

    Negotiated VC Cross-Connect Establishment

    (2a) The manager creates a row in this table by
        setting atmVcCrossConnectRowStatus to
        createAndWait(5).  The agent checks the
        requested topology and the mutual sanity of

```

the ATM traffic parameters and QoS Classes, i.e., the row creation fails if:

- the requested topology is not supported by the agent,
- the traffic/QoS parameter values associated with the requested row are incompatible with those of already existing rows for this VC cross-connect.

[For example, for setting up a point-to-point VC cross-connect, the ATM traffic parameters in the receive direction of a VCL at the low end of the cross-connect must equal to the traffic parameters in the transmit direction of the other VCL at the high end of the cross-connect, otherwise, the row creation fails.]  
The agent also checks for internal errors in building the cross-connect.

The atmVcCrossConnectIndex values in the corresponding atmVclTable rows are filled in by the agent at this point.

- (2b) The manager promotes the row in the atmVcCrossConnectTable by setting atmVcCrossConnectRowStatus to active(1). If this set is successful, the agent has reserved the resources specified by the ATM traffic parameter and QoS Class values for each direction of the VC cross-connect in an ATM switch or network.
- (3) The manager sets the atmVcCrossConnectAdminStatus to up(1) in all rows of this VC cross-connect to turn the traffic flow on.

#### One-Shot VC Cross-Connect Establishment

A VC cross-connect may also be established in one step by a set-request with all necessary parameter values and atmVcCrossConnectRowStatus set to createAndGo(4).

In contrast to the negotiated VC cross-connect establishment which allows for detailed error checking i.e., set errors are explicitly linked to

particular resource acquisition failures), the one-shot VC cross-connect establishment performs the setup on one operation but does not have the advantage of step-wise error checking.

#### VC Cross-Connect Retirement

A VC cross-connect identified by a particular value of `atmVcCrossConnectIndex` is released by:

- (1) Setting `atmVcCrossConnectRowStatus` of all rows identified by this value of `atmVcCrossConnectIndex` to `destroy(6)`. The agent may release all associated resources, and the `atmVcCrossConnectIndex` values in the corresponding `atmVclTable` row are removed. Note that a situation when only a subset of the associated rows are deleted corresponds to a VC topology change.
- (2) After deletion of the appropriate `atmVcCrossConnectEntries`, the manager may set `atmVclRowStatus` to `destroy(6)` the associated VCLs. The agent releases the resources and removes the associated rows in the `atmVclTable`.

#### VC Cross-Connect Reconfiguration

At the discretion of the agent, a VC cross-connect may be reconfigured by adding and/or deleting leafs to/from the VC topology as per the VC cross-connect establishment/retirement procedures. Reconfiguration of traffic/QoS parameter values requires release of the VC cross-connect before those parameter values may be changed for individual VCLs."

```
INDEX { atmVcCrossConnectIndex,
        atmVcCrossConnectLowIfIndex,
        atmVcCrossConnectLowVpi,
        atmVcCrossConnectLowVci,
        atmVcCrossConnectHighIfIndex,
        atmVcCrossConnectHighVpi,
        atmVcCrossConnectHighVci }
 ::= { atmVcCrossConnectTable 1 }
```



```

AtmVcCrossConnectEntry ::= SEQUENCE {
    atmVcCrossConnectIndex          INTEGER,
    atmVcCrossConnectLowIfIndex     IfIndex,
    atmVcCrossConnectLowVpi         INTEGER,
    atmVcCrossConnectLowVci         INTEGER,
    atmVcCrossConnectHighIfIndex    IfIndex,
    atmVcCrossConnectHighVpi        INTEGER,
    atmVcCrossConnectHighVci        INTEGER,
    atmVcCrossConnectAdminStatus     INTEGER,
    atmVcCrossConnectL2HOperStatus   INTEGER,
    atmVcCrossConnectH2LOperStatus   INTEGER,
    atmVcCrossConnectL2HLastChange   TimeStamp,
    atmVcCrossConnectH2LLastChange   TimeStamp,
    atmVcCrossConnectRowStatus       RowStatus
}

atmVcCrossConnectIndex OBJECT-TYPE
    SYNTAX  INTEGER (1..2147483647)
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
        "A unique value to identify this VC cross-connect."
    ::= { atmVcCrossConnectEntry 1 }

atmVcCrossConnectLowIfIndex OBJECT-TYPE
    SYNTAX  IfIndex
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
        "The value of this object is equal to MIB II's
        ifIndex value of the ATM interface port for this
        VC cross-connect. The term low implies
        that this ATM interface has the numerically lower
        ifIndex value than the other ATM interface
        identified in the same atmVcCrossConnectEntry."
    ::= { atmVcCrossConnectEntry 2 }

atmVcCrossConnectLowVpi OBJECT-TYPE
    SYNTAX  INTEGER (0..4095)
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
        "The value of this object is equal to the VPI
        value at the ATM interface
        associated with the VC cross-connect that is
        identified by atmVcCrossConnectLowIfIndex.
        The VPI value cannot exceed the number
        supported by the atmInterfaceMaxVpiBits"

```

at the low ATM interface port."  
 ::= { atmVcCrossConnectEntry 3 }

atmVcCrossConnectLowVci OBJECT-TYPE

SYNTAX INTEGER (0..65535)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The value of this object is equal to the VCI value at the ATM interface associated with this VC cross-connect that is identified by atmVcCrossConnectLowIfIndex. The VCI value cannot exceed the number supported by the atmInterfaceMaxVciBits at the low ATM interface port."

::= { atmVcCrossConnectEntry 4 }

atmVcCrossConnectHighIfIndex OBJECT-TYPE

SYNTAX IfIndex

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The value of this object is equal to MIB II's ifIndex value for the ATM interface port for this VC cross-connect. The term high implies that this VC cross-connect that this ATM interface has the numerically higher ifIndex value than the other ATM interface identified in the same atmVcCrossConnectEntry."

::= { atmVcCrossConnectEntry 5 }

atmVcCrossConnectHighVpi OBJECT-TYPE

SYNTAX INTEGER (0..4095)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The value of this object is equal to the VPI value at the ATM interface associated with the VC cross-connect that is identified by atmVcCrossConnectHighIfIndex. The VPI value cannot exceed the number supported by the atmInterfaceMaxVpiBits at the high ATM interface port."

::= { atmVcCrossConnectEntry 6 }

atmVcCrossConnectHighVci OBJECT-TYPE

SYNTAX INTEGER (0..65535)

MAX-ACCESS not-accessible

```

STATUS current
DESCRIPTION
  "The value of this object is equal to the VCI
  value at the ATM interface
  associated with the VC cross-connect that is
  identified by atmVcCrossConnectHighIfIndex.
  The VCI value cannot exceed
  the number supported by the atmInterfaceMaxVciBits
  at the high ATM interface port."
 ::= { atmVcCrossConnectEntry 7 }

```

```
atmVcCrossConnectAdminStatus OBJECT-TYPE
```

```

SYNTAX INTEGER {
    up(1),
    down(2)
}

```

```
MAX-ACCESS read-create
```

```
STATUS current
```

```
DESCRIPTION
```

```

"The value of this object identifies the desired
administrative status of this bi-directional
VC cross-connect. The up and down states
indicate that the traffic flow is enabled or
disabled respectively on this VC cross-connect."

```

```
DEFVAL { down }
```

```
::= { atmVcCrossConnectEntry 8 }
```

```
atmVcCrossConnectL2HOperStatus OBJECT-TYPE
```

```

SYNTAX INTEGER {
    up(1),
    down(2),
    unknown(3)
}

```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```

"The value of this object identifies the current
operational status of the VC cross-connect
in one direction; (i.e., from the low to
high direction). The up and down states indicate
that this ATM VC cross-connect from low
to high direction is operational or not
operational respectively. The unknown state
indicates that the state of it cannot be
determined."

```

```
::= { atmVcCrossConnectEntry 9 }
```

```

atmVcCrossConnectH2LOperStatus OBJECT-TYPE
    SYNTAX  INTEGER {
                up(1),
                down(2),
                unknown(3)
            }
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The value of this object identifies the current
        operational status of the VC cross-connect
        in one direction; (i.e., from the high to
        low direction).  The up and down states indicate
        that this ATM VC cross-connect from high
        to low direction is operational or not
        operational respectively.  The unknown state
        indicates that the state of it cannot be
        determined."
    ::= { atmVcCrossConnectEntry 10 }

atmVcCrossConnectL2HLastChange OBJECT-TYPE
    SYNTAX  TimeStamp
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The value of MIB II's sysUpTime object
        at the time this VC cross-connect entered
        its current operational state in low to high
        direction.  If the current state was
        entered prior to the last re-initialization of the
        agent, then this object contains a zero value."
    ::= { atmVcCrossConnectEntry 11 }

atmVcCrossConnectH2LLastChange OBJECT-TYPE
    SYNTAX  TimeStamp
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The value of MIB II's sysUpTime object
        at the time this VC cross-connect entered
        its current operational state in high to low
        direction.  If the current state was
        entered prior to the last re-initialization of the
        agent, then this object contains a zero value."
    ::= { atmVcCrossConnectEntry 12 }

atmVcCrossConnectRowStatus OBJECT-TYPE
    SYNTAX  RowStatus

```

```

MAX-ACCESS read-create
STATUS current
DESCRIPTION
  "The status of this entry in the
  atmVcCrossConnectTable. This object is used to
  create a new cross-connect for cross-connecting
  VCLs which are created using the atmVclTable
  or to change or delete existing cross-connect.
  This object must be initially set to
  'createAndWait' or 'createAndGo'. This object
  cannot be set to 'active' unless the following
  columnar object exists in this row:
  atmVcCrossConnectAdminStatus.
  To turn on a VC cross-connect,
  the atmVcCrossConnectAdminStatus
  is set to 'up'."
 ::= { atmVcCrossConnectEntry 13 }

```

```

-- AAL5 Virtual Channel Connection Performance Statistics
-- Group

-- This group contains the AAL5
-- performance statistics of a VCC at the
-- interface associated with an AAL5 entity in an ATM
-- host or ATM switch.

```

```

aal5VccTable OBJECT-TYPE
  SYNTAX SEQUENCE OF Aal5VccEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "This table contains AAL5 VCC performance
    parameters."
  ::= { atmMIBObjects 12 }

```

```

aal5VccEntry OBJECT-TYPE
  SYNTAX Aal5VccEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "This list contains the AAL5 VCC
    performance parameters."
  INDEX { ifIndex, aal5VccVpi, aal5VccVci }
  ::= { aal5VccTable 1 }

```

```

Aal5VccEntry ::= SEQUENCE {

```

```

aal5VccVpi                INTEGER,
aal5VccVci                INTEGER,
aal5VccCrcErrors          Counter32,
aal5VccSarTimeOuts       Counter32,
aal5VccOverSizedSDUs     Counter32
    }

```

```

aal5VccVpi                OBJECT-TYPE
    SYNTAX                 INTEGER (0..4095)
    MAX-ACCESS              not-accessible
    STATUS                  current
    DESCRIPTION
        "The VPI value of the AAL5 VCC at the
         interface identified by the ifIndex."
    ::= { aal5VccEntry 1 }

aal5VccVci                OBJECT-TYPE
    SYNTAX                 INTEGER (0..65535)
    MAX-ACCESS              not-accessible
    STATUS                  current
    DESCRIPTION
        "The VCI value of the AAL5 VCC at the
         interface identified by the ifIndex."
    ::= { aal5VccEntry 2 }

aal5VccCrcErrors          OBJECT-TYPE
    SYNTAX                 Counter32
    MAX-ACCESS              read-only
    STATUS                  current
    DESCRIPTION
        "The number of AAL5 CPCS PDUs received with
         CRC-32 errors on this AAL5 VCC at the
         interface associated with an AAL5 entity."
    ::= { aal5VccEntry 3 }

aal5VccSarTimeOuts       OBJECT-TYPE
    SYNTAX                 Counter32
    MAX-ACCESS              read-only
    STATUS                  current
    DESCRIPTION
        "The number of partially re-assembled AAL5
         CPCS PDUs which were discarded
         on this AAL5 VCC at the interface associated
         with an AAL5 entity because they
         were not fully re-assembled within the
         required time period.  If the re-assembly
         timer is not supported, then this object

```

```

        contains a zero value."
 ::= { aal5VccEntry 4 }

aal5VccOverSizedSDUs    OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS      current
    DESCRIPTION
        "The number of AAL5 CPCS PDUs discarded
         on this AAL5 VCC at the interface
         associated with an AAL5 entity because the
         AAL5 SDUs were too large."
 ::= { aal5VccEntry 5 }

-- Conformance Information

atmMIBConformance      OBJECT IDENTIFIER ::= { atmMIB 2 }

atmMIBGroups           OBJECT IDENTIFIER
                       ::= { atmMIBConformance 1 }
atmMIBCompliances     OBJECT IDENTIFIER
                       ::= { atmMIBConformance 2 }

-- Compliance Statements

atmMIBCompliance      MODULE-COMPLIANCE
    STATUS      current
    DESCRIPTION
        "The compliance statement for SNMP entities
         including networks which have ATM and
         AAL5 interfaces."

    MODULE -- this module
        MANDATORY-GROUPS {atmInterfaceConfGroup,
                           atmTrafficDescrGroup}

        OBJECT atmInterfaceMaxVpcs
        MIN-ACCESS read-only
        DESCRIPTION
            "Write access is not required."

        OBJECT atmInterfaceMaxVccs
        MIN-ACCESS read-only
        DESCRIPTION
            "Write access is not required."

```

OBJECT atmInterfaceMaxActiveVpiBits  
MIN-ACCESS read-only  
DESCRIPTION  
"Write access is not required."

OBJECT atmInterfaceMaxActiveVciBits  
MIN-ACCESS read-only  
DESCRIPTION  
"Write access is not required."

OBJECT atmInterfaceIlmiVpi  
MIN-ACCESS read-only  
DESCRIPTION  
"Write access is not required."

OBJECT atmInterfaceIlmiVci  
MIN-ACCESS read-only  
DESCRIPTION  
"Write access is not required."

OBJECT atmInterfaceMyNeighborIpAddress  
MIN-ACCESS read-only  
DESCRIPTION  
"Write access is not required."

OBJECT atmInterfaceMyNeighborIfName  
MIN-ACCESS read-only  
DESCRIPTION  
"Write access is not required."

OBJECT atmTrafficDescrType  
MIN-ACCESS read-only  
DESCRIPTION  
"Write access is not required."

OBJECT atmTrafficDescrParam1  
MIN-ACCESS read-only  
DESCRIPTION  
"Write access is not required."

OBJECT atmTrafficDescrParam2  
MIN-ACCESS read-only  
DESCRIPTION  
"Write access is not required."

OBJECT atmTrafficDescrParam3  
MIN-ACCESS read-only  
DESCRIPTION



"Write access is not required."

OBJECT atmTrafficDescrParam4  
 MIN-ACCESS read-only  
 DESCRIPTION  
 "Write access is not required."

OBJECT atmTrafficDescrParam5  
 MIN-ACCESS read-only  
 DESCRIPTION  
 "Write access is not required."

OBJECT atmTrafficQoSClass  
 MIN-ACCESS read-only  
 DESCRIPTION  
 "Write access is not required."

OBJECT atmTrafficDescrRowStatus  
 SYNTAX INTEGER {active(1)}  
 -- subset of RowStatus  
 MIN-ACCESS read-only  
 DESCRIPTION  
 "Write access is not required, and only one  
 of the six enumerated values for the  
 RowStatus textual convention need be  
 supported, specifically: active(1)."

GROUP atmInterfaceDs3PlcpGroup  
 DESCRIPTION  
 "This group is mandatory only for those  
 ATM interfaces which implement the  
 DS3 PLCP layer."

GROUP atmInterfaceTCGroup  
 DESCRIPTION  
 "This group is mandatory only for those  
 ATM interfaces which implement the  
 TC Sublayer."

GROUP atmVpcTerminationGroup  
 DESCRIPTION  
 "This group is mandatory only for those  
 ATM interfaces which implement ATM  
 VPLs that terminate VPCs (i.e., ones which  
 are NOT cross-connected to other VPLs)."

GROUP atmVpCrossConnectGroup  
 DESCRIPTION

"This group is mandatory only for those ATM interfaces which implement ATM VPLs that are not associated with VCLs and are cross-connected to other VPLs."

OBJECT atmVplVpi  
SYNTAX INTEGER (1..255)  
DESCRIPTION  
"For ATM UNIs supporting VPCs, the VPI value ranges from 1 to 255."

OBJECT atmVplAdminStatus  
MIN-ACCESS read-only  
DESCRIPTION  
"Write access is not required."

OBJECT atmVplReceiveTrafficDescrIndex  
MIN-ACCESS read-only  
DESCRIPTION  
"Write access is not required."

OBJECT atmVplTransmitTrafficDescrIndex  
MIN-ACCESS read-only  
DESCRIPTION  
"Write access is not required."

OBJECT atmVplRowStatus  
SYNTAX INTEGER {active(1)}  
-- subset of RowStatus  
MIN-ACCESS read-only  
DESCRIPTION  
"Write access is not required, and only one of the six enumerated values for the RowStatus textual convention need be supported, specifically: active(1)."

OBJECT atmVpCrossConnectLowVpi  
SYNTAX INTEGER (1..255)  
DESCRIPTION  
"For ATM UNIs supporting VPCs, the VPI value at the numerically lower ATM interface port index number ranges from 1 to 255."

OBJECT atmVpCrossConnectHighVpi  
SYNTAX INTEGER (1..255)  
DESCRIPTION  
"For ATM UNIs supporting VPCs, the VPI value at the numerically higher ATM interface port

index number ranges from 1 to 255."

OBJECT atmVpCrossConnectAdminStatus  
MIN-ACCESS read-only  
DESCRIPTION

"Write access is not required."

OBJECT atmVpCrossConnectRowStatus  
SYNTAX INTEGER {active(1)}  
-- subset of RowStatus  
MIN-ACCESS read-only  
DESCRIPTION

"Write access is not required, and only one of the six enumerated values for the RowStatus textual convention need be supported, specifically: active(1)."

GROUP atmVccTerminationGroup  
DESCRIPTION

"This group is mandatory only for those ATM interfaces which implement ATM VCLs that terminate VCCs (i.e., ones which are NOT cross-connected to other VCLs)."

GROUP atmVcCrossConnectGroup  
DESCRIPTION

"This group is mandatory only for those ATM interfaces which implement ATM VCLs that are cross-connected to other VCLs."

OBJECT atmVclVpi  
SYNTAX INTEGER (0..255)  
DESCRIPTION

"For ATM UNIs supporting VCCs, the VPI value ranges from 0 to 255."

OBJECT atmVclAdminStatus  
MIN-ACCESS read-only  
DESCRIPTION

"Write access is not required."

OBJECT atmVclReceiveTrafficDescrIndex  
MIN-ACCESS read-only  
DESCRIPTION

"Write access is not required."

OBJECT atmVclTransmitTrafficDescrIndex

MIN-ACCESS read-only  
DESCRIPTION  
"Write access is not required."

OBJECT atmVccAalType  
MIN-ACCESS read-only  
DESCRIPTION  
"Write access is not required."

OBJECT atmVclRowStatus  
SYNTAX INTEGER {active(1)}  
-- subset of RowStatus  
MIN-ACCESS read-only  
DESCRIPTION  
"Write access is not required, and only one  
of the six enumerated values for the  
RowStatus textual convention need be  
supported, specifically: active(1)."

OBJECT atmVcCrossConnectLowVpi  
SYNTAX INTEGER (0..255)  
DESCRIPTION  
"For ATM UNIs supporting VCCs, the VPI value  
at the numerically lower ATM interface port  
index number ranges from 0 to 255."

OBJECT atmVcCrossConnectHighVpi  
SYNTAX INTEGER (0..255)  
DESCRIPTION  
"For ATM UNIs supporting VCCs, the VPI value  
at the numerically higher ATM interface port  
index number ranges from 0 to 255."

OBJECT atmVcCrossConnectAdminStatus  
MIN-ACCESS read-only  
DESCRIPTION  
"Write access is not required."

OBJECT atmVcCrossConnectRowStatus  
SYNTAX INTEGER { active(1)}  
-- subset of RowStatus  
MIN-ACCESS read-only  
DESCRIPTION  
"Write access is not required, and only one  
of the six enumerated values for the  
RowStatus textual convention need be  
supported, specifically: active(1)."

```

GROUP    aal5VccGroup
DESCRIPTION
    "This group is mandatory for the
    AAL5 virtual connections only."

OBJECT    aal5VccVpi
SYNTAX    INTEGER (0..255)
DESCRIPTION
    "For ATM UNIs supporting AAL5 VCCs,
    the VPI value ranges from 0 to 255."

OBJECT    atmVccAal5CpcsTransmitSduSize
MIN-ACCESS read-only
DESCRIPTION
    "Write access is not required."

OBJECT    atmVccAal5CpcsReceiveSduSize
MIN-ACCESS read-only
DESCRIPTION
    "Write access is not required."

OBJECT    atmVccAal5EncapsType
MIN-ACCESS read-only
DESCRIPTION
    "Write access is not required."
    ::= { atmMIBCompliances 1 }

```

-- Units of Conformance

```

atmInterfaceConfGroup    OBJECT-GROUP
    OBJECTS {
        atmInterfaceMaxVpcs, atmInterfaceMaxVccs,
        atmInterfaceConfVpcs, atmInterfaceConfVccs,
        atmInterfaceMaxActiveVpiBits,
        atmInterfaceMaxActiveVciBits,
        atmInterfaceIlmiVpi,
        atmInterfaceIlmiVci,
        atmInterfaceAddressType,
        atmInterfaceAdminAddress,
        atmInterfaceMyNeighborIpAddress,
        atmInterfaceMyNeighborIfName}
    STATUS    current
    DESCRIPTION
        "A collection of objects providing configuration
        information about an ATM interface."
    ::= { atmMIBGroups 1 }

```

```

atmTrafficDescrGroup    OBJECT-GROUP
  OBJECTS {
    atmTrafficDescrType, atmTrafficDescrParam1,
    atmTrafficDescrParam2, atmTrafficDescrParam3,
    atmTrafficDescrParam4, atmTrafficDescrParam5,
    atmTrafficQoSClass, atmTrafficDescrRowStatus}
  STATUS      current
  DESCRIPTION
    "A collection of objects providing information
    about ATM traffic descriptor type and
    the associated parameters."
  ::= { atmMIBGroups 2 }

atmInterfaceDs3PlcpGroup  OBJECT-GROUP
  OBJECTS {atmInterfaceDs3PlcpSEFSs,
    atmInterfaceDs3PlcpAlarmState,
    atmInterfaceDs3PlcpUASs}
  STATUS      current
  DESCRIPTION
    "A collection of objects providing information
    about DS3 PLCP layer at an ATM interface."
  ::= { atmMIBGroups 3 }

atmInterfaceTCGroup      OBJECT-GROUP
  OBJECTS { atmInterfaceOCDEvents,
    atmInterfaceTCAlarmState }
  STATUS      current
  DESCRIPTION
    "A collection of objects providing information
    about TC sublayer at an ATM interface."
  ::= { atmMIBGroups 4 }

atmVpcTerminationGroup   OBJECT-GROUP
  OBJECTS {atmVplOperStatus, atmVplAdminStatus,
    atmVplLastChange,
    atmVplReceiveTrafficDescrIndex,
    atmVplTransmitTrafficDescrIndex,
    atmVplRowStatus }
  STATUS      current
  DESCRIPTION
    "A collection of objects providing
    information about a VPL at an ATM interface
    which terminates a VPC
    (i.e., one which is NOT cross-connected
    to other VPLs)."
  ::= { atmMIBGroups 5 }

atmVccTerminationGroup   OBJECT-GROUP

```

```

OBJECTS { atmVclOperStatus, atmVclAdminStatus,
          atmVclLastChange,
          atmVclReceiveTrafficDescrIndex,
          atmVclTransmitTrafficDescrIndex,
          atmVccAalType, atmVclRowStatus }
STATUS      current
DESCRIPTION
  "A collection of objects providing information
  about a VCL at an ATM interface
  which terminates a VCC (i.e., one which is
  NOT cross-connected to other VCLs)."
 ::= { atmMIBGroups 6 }

```

```

atmVpCrossConnectGroup    OBJECT-GROUP
OBJECTS { atmVplReceiveTrafficDescrIndex,
          atmVplTransmitTrafficDescrIndex,
          atmVplOperStatus, atmVplRowStatus,
          atmVpCrossConnectAdminStatus,
          atmVpCrossConnectL2HOperStatus,
          atmVpCrossConnectH2LOperStatus,
          atmVpCrossConnectL2HLastChange,
          atmVpCrossConnectH2LLastChange,
          atmVpCrossConnectRowStatus,
          atmVplCrossConnectIdentifier,
          atmVpCrossConnectIndexNext }
STATUS      current
DESCRIPTION
  "A collection of objects providing
  information about a VP cross-connect
  and the associated VPLs that are
  cross-connected together."
 ::= { atmMIBGroups 7 }

```

```

atmVcCrossConnectGroup    OBJECT-GROUP
OBJECTS { atmVclReceiveTrafficDescrIndex,
          atmVclTransmitTrafficDescrIndex,
          atmVclOperStatus, atmVclRowStatus,
          atmVcCrossConnectAdminStatus,
          atmVcCrossConnectL2HOperStatus,
          atmVcCrossConnectH2LOperStatus,
          atmVcCrossConnectL2HLastChange,
          atmVcCrossConnectH2LLastChange,
          atmVcCrossConnectRowStatus,
          atmVclCrossConnectIdentifier,
          atmVcCrossConnectIndexNext }
STATUS      current
DESCRIPTION
  "A collection of objects providing

```

```

        information about a VC cross-connect
        and the associated VCLs that are
        cross-connected together."
 ::= { atmMIBGroups 8 }

aal5VccGroup    OBJECT-GROUP
OBJECTS {atmVccAal5CpcsTransmitSduSize,
        atmVccAal5CpcsReceiveSduSize,
        atmVccAal5EncapsType,
        aal5VccCrcErrors, aal5VccSarTimeOuts,
        aal5VccOverSizedSDUs }
STATUS        current
DESCRIPTION
    "A collection of objects providing
    AAL5 configuration and performance statistics
    of a VCC."
 ::= { atmMIBGroups 9 }

END

```

## 11. Acknowledgments

This memo is the result of the work of the AToMMIB Working Group. In particular, the contributions of Keith McCloghrie and Ted Brunner were critical to the formulation of this specification.

## 12. References

- [1] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1442, SNMP Research, Inc., Hughes LAN Systems, Dover Beach Consulting, Inc., Carnegie Mellon University, April 1993.
- [2] McCloghrie, K., and M. Rose, Editors, "Management Information Base for Network Management of TCP/IP-based internets: MIB-II", STD 17, RFC 1213, Hughes LAN Systems, Performance Systems International, March 1991.
- [3] Galvin, J., and K. McCloghrie, "Administrative Model for version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1445, Trusted Information Systems, Hughes LAN Systems, April 1993.
- [4] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Protocol Operations for version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1448, SNMP Research, Inc., Hughes LAN Systems, Dover Beach Consulting, Inc., Carnegie Mellon University, April 1993.



- [5] McCloghrie, K. and F. Kastenholz, "Evolution of the Interfaces Group of MIB-II", RFC 1573, Hughes LAN Systems, FTP Software, January 1994.
- [6] Brown, T., and K. Tesink, "Definitions of Managed Objects for SMDS Interfaces", RFC 1694, Bellcore, August 1994.
- [7] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Textual Conventions for SNMPv2", RFC 1443, SNMP Research, Inc., Hughes LAN Systems, Dover Beach Consulting, Inc., Carnegie Mellon University, April 1993.
- [8] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Conformance Statements for SNMPv2", RFC 1444, SNMP Research, Inc., Hughes LAN Systems, Dover Beach Consulting, Inc., Carnegie Mellon University, April 1993.
- [9] "ATM Forum UNI Specification, Version 3.0", September 1993.
- [10] "ATM Forum B-ICI Specification, Version 1.0", September 1993.

### 13. Security Considerations

Security issues are not discussed in this memo.

### 14. Authors' Addresses

Masuma Ahmed  
Bell Communications Research  
Room 1F-217  
331 Newman Springs Road  
P.O. Box 7020  
Red Bank, NJ 07701-7020

Phone: (908) 758-2515  
EMail: mxa@mail.bellcore.com

Kaj Tesink  
Bell Communications Research  
Room 1A-427  
331 Newman Springs Road  
P.O. Box 7020  
Red Bank, NJ 07701-7020

Phone: (908) 758-5254  
EMail: kaj@cc.bellcore.com

