Section 4:

Interim Local Management Interface Specification

Scope

Whereas the ITU-T and ANSI standards committees have been working to define both C-plane and U-plane procedures for ATM, local network management procedures in the M-plane are in large part considered "for further study".

In the interim period until such standards are available, the Simple Network Management Protocol (SNMP) and an ATM UNI Management Information Base (MIB) will be required to provide any ATM user device with status and configuration information concerning the Virtual Path and Channel Connections available at its UNI. In addition, more global operations and network management information (e.g. status and operational measurement information for the public and private UNI as defined in this document) may also facilitate diagnostics procedures at the UNI.

The ILMI fits into the overall management model for an ATM device as illustrated in Figure 4-1 as clarified by the following principles and options.

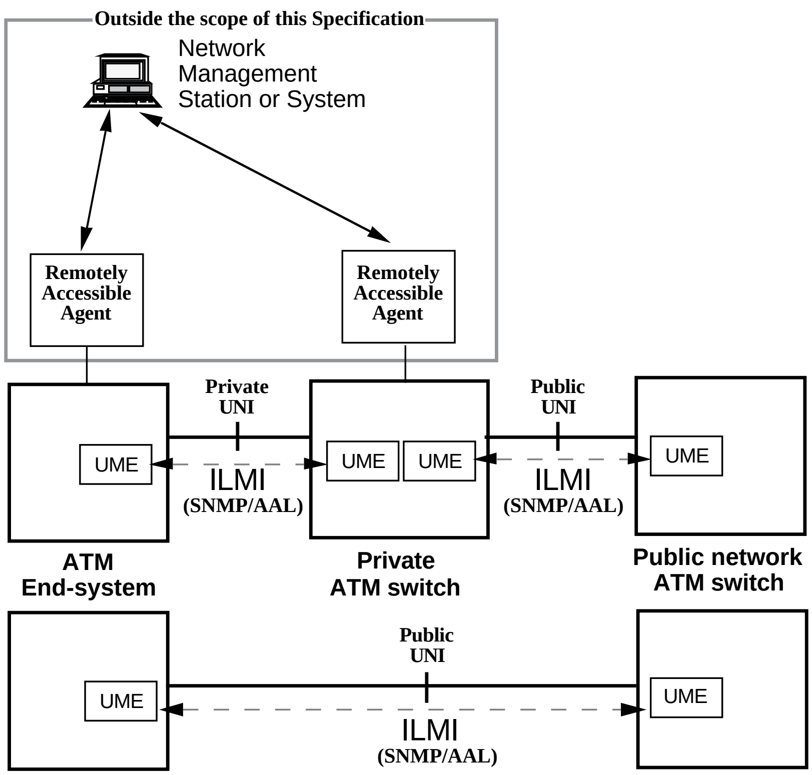

Figure 4-1 Definition and Context of ILMI

- Each ATM device shall support one or more UNIs.
- Interim Local Management Interface (ILMI) functions for a UNI provide status, configuration, and control information about link and physical layer parameters at the UNI.
- ILMI functions for a UNI also provide for address registration across the UNI. Further details on address registration are contained in section 5.8.
- There is a per-UNI set of managed objects, the UNI ILMI attributes, that is sufficient to support the ILMI functions for each UNI.
- The UNI ILMI attributes are organized in a standard MIB structure; there is one UNI ILMI MIB structure instance for each UNI.
- There is one MIB instance per ATM device, which contains one or more UNI ILMI MIB structures. This supports the need for general network management systems to have access to the information in the UNI ILMI MIB structures.
- For any ATM device, there is a UNI Management Entity (UME) associated with each UNI that supports the ILMI functions for that UNI, including coordination between the physical and ATM layer management entities associated with that UNI.
- When two ATM devices are connected across a *(point-to-point)* UNI, there are two UNI management entities (UMEs) associated with that UNI, one UME for each ATM device, and two such UMEs are defined as adjacent UMEs.
- The ILMI communication takes place between adjacent ATM UMEs.
- The ILMI communication protocol is an open management protocol (i.e., SNMP/AAL initially).
- A UNI Management Entity (UME) can access, via the ILMI communication protocol, the UNI ILMI MIB information associated with its adjacent UME.
- Whether access to additional information (beyond the adjacent UME's UNI ILMI MIB information) is available via the ILMI communication protocol is currently unspecified, and is regarded as a vendor implementation choice.
- Separation of the MIB structure from the access methods allows for the use of multiple access methods for management information. For the ILMI function, the access method is an open management protocol (i.e., SNMP/AAL) over a well known VPI/VCI value. For example, general network management applications (e.g., from a Network Management Station (NMS) performing generic Customer Network Management (CNM) functions) the access method is also an open management protocol (e.g., SNMP/UDP/IP/AAL) over a specific VPI/VCI value (or a completely separate communications method)

allocated to support the general management applications. The peer entity in an ATM device that communicates directly with a NMS is a management agent, not a UME; however, since the management agent can access the MIB instance for the ATM device it can access all of the UNI ILMI MIB structure instances.

- This document pertains to the UNI ILMI MIB structure of the “Local UNI” (i.e., between adjacent UMEs) only.

The Simple Network Management Protocol (SNMP) without UDP and IP addressing along with an ATM UNI Management Information Base (MIB) were chosen for the ILMI.

4.1 Interim Local Management Interface (ILMI) Functions

An Interim Local Management Interface (ILMI) supports bi-directional exchange of management information between UNI Management Entities (UMEs) related to UNI ATM layer and physical layer parameters. The communication across the ILMI is protocol symmetric. In addition, each of the adjacent UMEs supporting ILMI will contain an agent application and may contain a management application. Unless otherwise stated for specific portions of the MIB, both of the adjacent UMEs contain the same Management Information Base (MIB). However, semantics of some MIB objects may be interpreted differently. As shown in Figure 4-2, an example list of the equipment that will use the ATM UNI ILMI include:

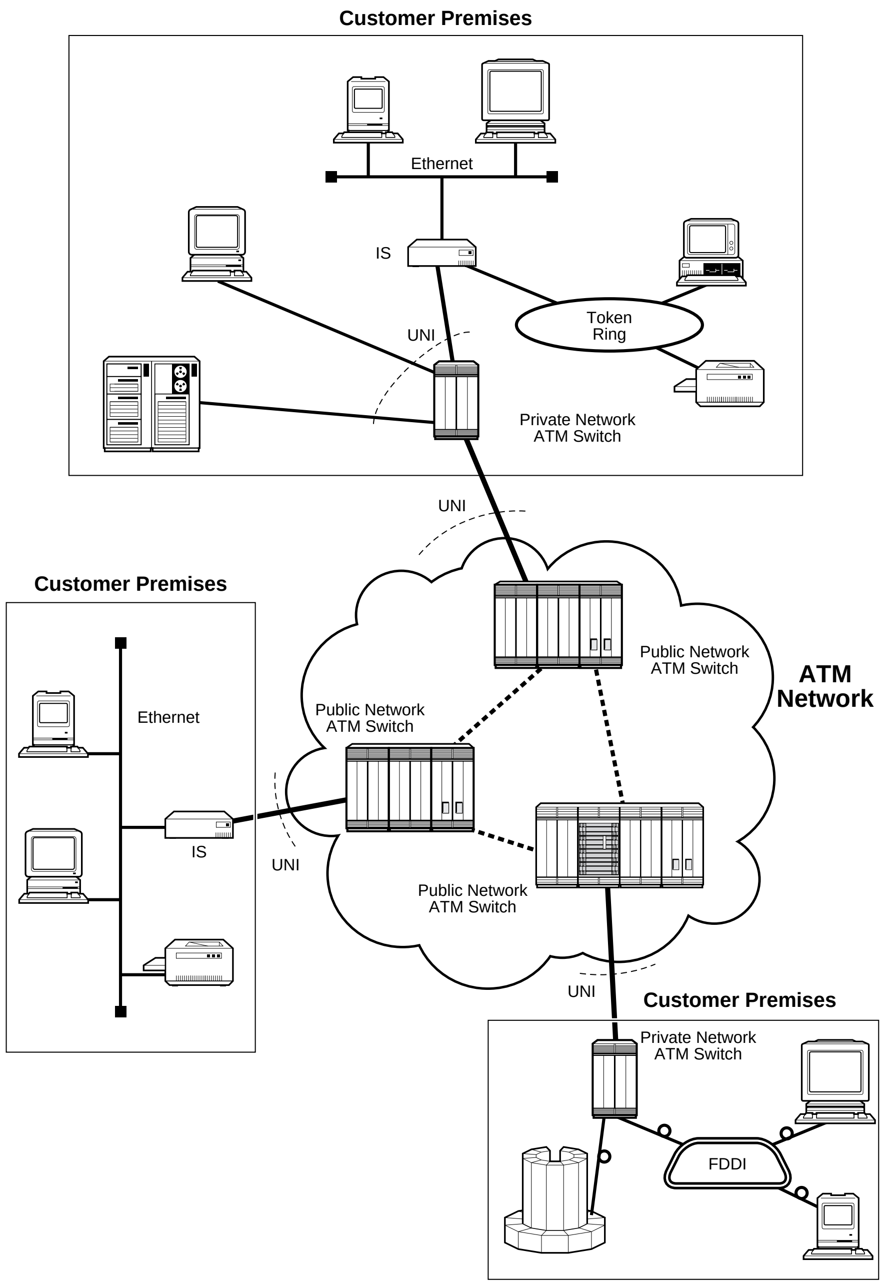

Figure 4-2 Examples of Equipment Implementing the ATM UNI ILMI

- Higher layer switches such as internet routers,frame relay switches, or LAN bridges, that transfer their frames within ATM cells and forward the cells across an ATM UNI to an ATM switch.

- Workstations and computers with ATM interfaces which send their data in ATM cells across an ATM UNI to an ATM switch.

- ATM Network switches which send ATM cells across an ATM UNI to other ATM devices.

4.2 ILMI Service Interface

The Interim Local Management Interface uses SNMP for monitoring and control operations of ATM management information across the UNI. The ATM UNI management information will be represented in a Management Information Base. The types of management information will be available in the ATM UNI MIB are as follows:

- Physical Layer
- ATM Layer
- ATM Layer Statistics
- Virtual Path (VP) Connections
- Virtual Channel (VC) Connections
- Address Registration Information

The tree structure of the ATM UNI ILMI MIB is depicted in Figure 4-3.

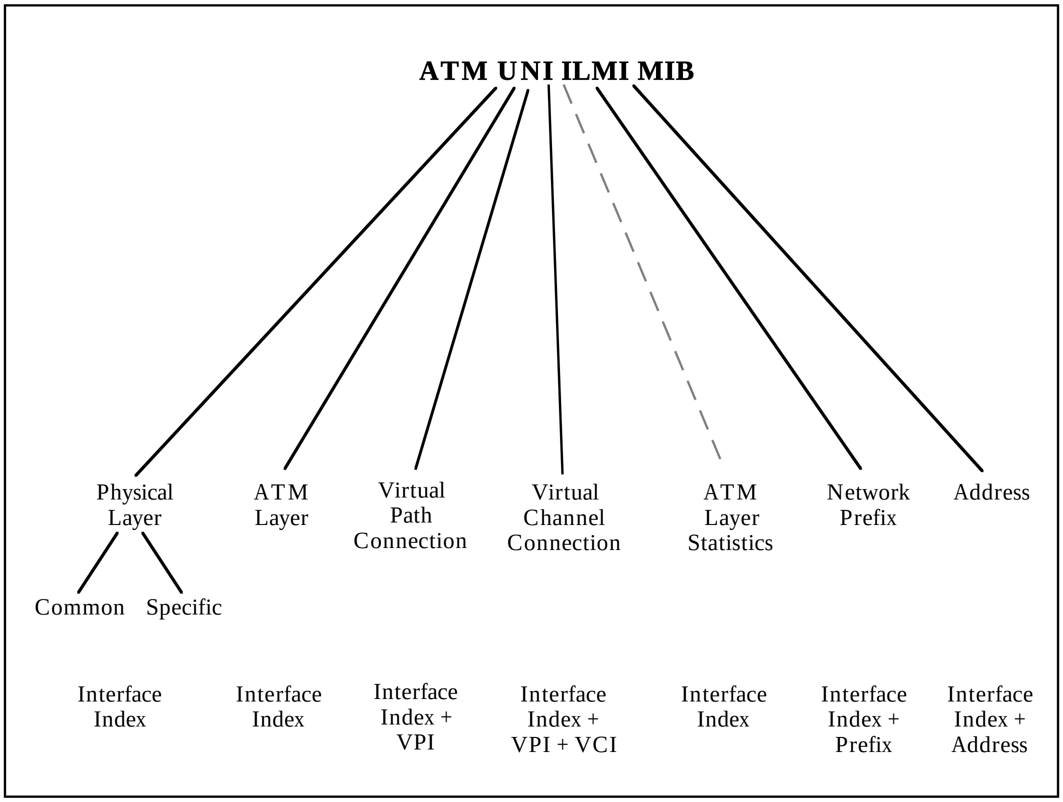

Figure 4-3 ILMI ATM UNI MIB Tree Structure

The ATM UNI MIB may be extended over time to allow for the addition of new items without requiring any changes to the management protocol or framework. In addition, vendors can define private ATM UNI MIB extensions to support additional or proprietary features of their products.

The following sections introduce the groups which categorize the management information. An entire tree group is either Optional **(O),** Conditionally Required **(CR)** or Required **(R)**. If a group is Required, then every element in the group is Required. For a group which is Conditionally Required it follows that it every element in the group is required if implemented.

4.2.1 System (R)

The System group from RFC 1213 is included to define the following objects as defined in section 4.5.

4.2.2 Physical Layer (R)

The physical layer interfaces which the ILMI supports is identified as defined in section 2.

The physical media which the ILMI supports is identified as defined in section 2.

Each physical interface type has a set of specific attributes and information associated with it.

Status information reflects the state of the physical link connecting the adjacent UMEs.

4.2.3 ATM Layer (R)

The ILMI provides access to management information about the ATM Layer as defined in section 3.1.

There is one ATM layer per physical interface.

Certain attributes of the ATM layer are common across all Virtual Path Connections (VPCs) and Virtual Channel Connections (VCCs) at this UNI.

Configuration information at the ATM layer relates to the size of the VPI and VCI address fields in the ATM cell header, number of configured permanent VPCs and permanent VCCs, and maximum number of VPCs and VCCs allowed at this UNI.

4.2.4 ATM Layer Statistics (O)

Certain objects and attributes represent aggregate statistics across all VPCs and VCCs on the Local ATM UNI, accumulated over time.

4.2.5 Virtual Path Connections (R)

In the context of supporting ILMI functions, a point-to-point Virtual Path Connection (VPC) extends between two ATM User-Network Interfaces that terminate the VPC.

On the local ATM Layer interface the VPC is uniquely identified by the VPI value.

The status information indicates the UME's knowledge of the VPC status. The VPC status may be either end-to-end, local or unknown.

Configuration information relates to the QoS parameters for the VPC local end point.

4.2.6 Virtual Channel Connections (R)

In the context of supporting ILMI functions, a point-to-point Virtual Channel Connection (VCC) extends between two ATM User-Network Interfaces that terminate the VCC.

On the local ATM Layer interface the VCC is uniquely identified by the VPI and VCI value.

The status information indicates the UME's knowledge of the VCC status. The VCC status may be either end-to-end, local or unknown.

Configuration information relates to the QoS parameters for the VPC local end point.

4.3 Simple Network Management Protocol (SNMP)

The SNMP protocol as defined in RFC 1157 consists of four types of operations which are used to manipulate management information. These are:

Get	Used to retrieve specific management information.
Get-Next	Used to retrieve, via traversal of the MIB, management information.
Set	Used to alter management information.
Trap	Used to report extraordinary events.

4.4 Management Information Base (MIB) Model for ILMI Managed Objects

Information related to the operation of the ATM UNI is organized into a MIB in a hierarchical fashion as defined in section 4.2. Each ATM UNI corresponds to just one physical interface. Logically, the ATM UNI MIB accessed over the ILMI corresponds to the single ATM UNI/physical interface. Implementations of devices which have multiple physical interfaces (e.g., multiple interface end user devices, private switches and public switches) may implement a single ATM UNI MIB, indexed for each ATM UNI/physical interface.

The MIB attributes are used in the standard fashion: all attributes are by default read only across the ILMI, unless stated otherwise as readable and writeable across the ILMI for a specific UNI MIB variable.

The ILMI MIB for ATM devices is specified under the ATM Forum's node under the enterprises node of the standard SMI as defined in RFC 1212. This means it is prefixed by the following tree path or name, 1.3.6.1.4.1.353.

4.4.1 Per-System UNI MIB Attributes (R)

No per-system object is defined in the ILMI MIB. UMEs implementing the ILMI MIB are also expected to support the "system" group defined in MIB-II, see section 4.5.

4.4.2 Per-Physical Interface UNI MIB Attributes (R)

These attributes are located in the Physical Port Group (atmfPhysicalGroup). The physical interface is identified by the interface index (atmfPortIndex). MIB information at this level includes:

Common

- Interface Index
- Interface Address
- Transmission Type
- Media Type
- Operational Status
- Adjacency information

Transmission Type Specific Information

4.4.2.1 Common (R)

Certain objects and attributes are common to all transmission types.

4.4.2.1.1 Interface Index (R)

The interface index is used in the ATM Layer, VPC and VCC ATM UNI MIB subtrees to identify a particular physical interface on the ATM device.

When this object has the value zero, it implicitly identifies the physical interface over which ILMI messages are received. Only implicit identification is mandatory.

Optionally, one of many interfaces may be explicitly identified by an interface index unique to the ATM device.

4.4.2.1.2 Interface Address (O)

The Interface Address object specified in version 2.0 of the ATM Forum UNI Specification is deprecated. The Address Registration extensions, specified in section 5.8 of this specification, replaces the Interface Address object. The Interface Address object should not be implemented except as required for backward compatibility.

4.4.2.1.3 Transmission Type (R)

The following transmission types are currently supported, as defined in the indicated section

2.1 SONET STS-3c Physical Layer Interface
2.2 DS3 Physical Layer Interface
2.3 Physical Layer for 100 Mb/s Interface
2.4 Physical Layer for 155 Mb/s Interface

The Transmission Speed (in units of cells/second) is uniquely determined by the Transmission Type as defined in section 2.

4.4.2.1.4 Media Type (R)

Physical media types defined in the MIB currently are:

- Coaxial Cable
- Single Mode Fiber
- Multi-Mode Fiber
- Shielded Twisted Pair
- Unshielded Twisted Pair

4.4.2.1.5 Physical Layer Operational Status (R)

This object allows the adjacent UMEs to observe the operational state of the physical interface supported between them. Valid values are:

- In-Service
- Out-of-Service
- Loop Back Mode

If the Operational Status has the value of Out-of-Service, then the ILMI should not alarm on the physical interface. This capability is useful if equipment is to be disconnected, or for troubleshooting purposes.

The Loop Back Mode attribute indicates that a local loopback in place. For example C-Bit parity DS3 has a standard method of commanding a local interface loopback using physical layer signalling.

4.4.2.1.6 Adjacency information

This information allows the neighboring system to maintain a table of adjacent systems to facilitate autodiscovery and tracing of ATM connections by Network Management Systems. It includes the address to which a Network Management Station can send Network Management protocol messages, and the name of the interface.

4.4.2.2 Transmission Type Specific Information (R)

This object is a reference to any additional management information which is available for the physical interaces and which is specific to the interface's transmission type.

4.4.2.2.1 SONET STS-3c Physical Layer Interface

Currently undefined.

4.4.2.2.2 DS3 Physical Layer Interface

This is defined as a reference to RFC 1407.

4.4.2.2.3 Physical Layer for 100 Mbps Interface

Currently undefined.

4.4.2.2.4 Physical Layer for 155 Mbps Interface

Currently undefined.

4.4.3 Per-ATM Layer Interface UNI MIB Attributes (R)

These attributes are located in the ATM Layer Group (atmfATMLayerGroup). The interface is identified by the interface index (atmfAtmLayerIndex). MIB information at this level includes:

Interface Index

Configuration Information

- Maximum Number of VPCs
- Maximum Number of VCCs
- VPI/VCI Address Width
- Number of Configured VPCs
- Number of Configured VCCs
- ATM UNI Port Type (Public/Private)

4.4.3.1 Interface Index (R)

The Interface Index is the same as that for the Physical interface as defined in section 4.4.2.1.

It is implicitly the local ATM UNI, unless the optional mode of explicit identification is supported.

4.4.3.2 Configuration Information (R)

Configuration information about the number of Virtual Path Connections (VPCs) and Virtual Channel Connections (VCCs) on the local interface is defined here.

4.4.3.2.1 Maximum Number of VPCs (R)

This is the maximum number of switched and permanent VPCs which the local interface can support.

4.4.3.2.2 Maximum Number of VCCs (R)

This is the maximum number of switched and permanent VCCs which the local interface can support.

4.4.3.2.3 Number of Configured VPCs (R)

This is the current number of permanentVPCs for which the local interface is configured to process. This number represents the number of entries in the atmfVpcTable.

4.4.3.2.4 Number of Configured VCCs (R)

This is the current number of permanent VCCs for which the local interface is configured to process. This number represents the number of entries in the atmfVpcTable.

4.4.3.2.5 Maximum Number of Active VPI bits (R)

This is the maximum number of VPI bits that can be active for this interface.

4.4.3.2.6 Maximum Number of Active VCI bits (R)

This is the maximum number of VCI bits that can be active for this interface.

4.4.3.2.7 ATM UNI Port Type (R)

This parameter indicates whether the ATM UNI is of the public or private type.

4.4.3.2.8 ATM UNI Version (R)

This is the latest version of the UNI specification supported on this UNI.

If the peer UME's value of this object is the same as, or later than the local UME's value, then the version corresponding to the local UME's value should be attempted. Otherwise, if the peer UME's value of this object is earlier, and supported locally, then the local UME should attempt the version corresponding to the peer UME's value. Otherwise, compatibility of the two UMEs cannot be assumed.

4.4.4 ATM Layer Statistics (O)

These attributes are located in the ATM Statistics Group (atmfStatisticsGroup). This group is indexed by the interface index (atmfAtmStatsIndex). MIB information at this level includes:

- Interface Index
- ATM Cells Received
- ATM Cells Dropped on the Receive Side
- ATM Cells Transmitted

Certain counters make sense only at the ATM Layer for an entire physical interface, while others are rolled up across all VPCs and VCCs at an interface. These counters are thus aggregated across the entire ATM UNI, and accumulated over time. All counters are 32 bits which wrap around.

The optional ATM layer statistics can be used for the following purposes.

- Identify problems that affect UNI performance, such as:
  - Loss due to bit errors, and/or
  - Unavailability of link connections.
- Aid in problem diagnosis and troubleshooting.
- Collect traffic engineering data.

4.4.4.1 ATM Cells Received (O)

This is a count of the number of ATM cells received across the ATM UNI which were assigned and not dropped.

4.4.4.2 ATM Cells Dropped on the Receive Side (O)

This is a count of the number of ATM cells received across the ATM UNI which were dropped for the following specific reasons. The reasons for which an ATM cell can increment this counter shall be:

The Header Error Control (HEC) processing either detected an error, or identified an uncorrectable error once cell delineation has been achieved.

An invalidly formatted cell header (as defined in section 3.4.3) was received.

This device was not configured to process the received VPI/VCI (i.e., it is unknown). For example, a switch translation table entry was not defined for the received VPI/VCI value.

4.4.4.3 ATM Cells Transmitted (O)

This is a count of the number of ATM cells transmitted across the ATM UNI which were assigned. This is a count of cells carrying actual user data across the ATM UNI.

4.4.5 Per-Virtual Path UNI MIB Attributes (R)

These attributes are located in the Virtual Path Group (atmfVpcGroup). This group is indexed by the interface index (atmfVpcPortIndex) and the VPI value (atmfVpcVpi). Only permanent virtual path connections are represented in this group. MIB information at this level includes:

- Interface Index
- VPI Value
- Transmit Traffic Descriptor
- Receive Traffic Descriptor
- Operational Status
- Transmit QoS Class
- Receive QoS Class

4.4.5.1 Interface Index (R)

The Interface Index is the same as that for the Physical interface as defined in section 4.4.2.1.1

It is implicitly the local ATM UNI, unless the optional mode of explicit identification is supported.

4.4.5.2 VPI (R)

This is the value of the Virtual Path Identifier (VPI) for this VPC.

4.4.5.3 Transmit Traffic Descriptor (R)

This is a specification of the conformance definition and associated source traffic descriptor parameter values described in section 3.6.2.5 that are applicable to the transmit side of this interface for this VPC.

4.4.5.4 Receive Traffic Descriptor (R)

This is a specification of the conformance definition and associated source traffic descriptor parameter values described in section 3.6.2.5 that are applicable to the receive side of this interface for this VPC.

4.4.5.5 Operational Status (R)

This object represents the state of the VPC as known by the local device. If the end-to-end status is known then a value of end2endUp(2) or end2endDown(3) is used. If only the local status is known then a value of localUpEnd2endUnknown(4) or localDown(5) is used.

4.4.5.6 Transmit (QoS) Class (R)

This is the QoS Class defined in section 4 of Appendix A, for the transmit side of this interface for thisVPC.

4.4.5.6 Receive (QoS) Class (R)

This is the QoS Class defined in section 4 of Appendix A, for the receive side of this interface for thisVPC.

4.4.6 Per-Virtual Channel UNI MIB Attributes (R)

These attributes are located in the Virtual Channel Group (atmfVccGroup). This group is indexed by the interface index (atmfVccPortIndex), VCC VPI value (atmfVccVpi) and VCC VCI value (atmfVccVci). Only permanent virtual channel connections are represented in this group. MIB information at this level includes:

- Interface Index
- VPI/VCI Value
- Transmit Traffic Descriptor
- Receive Traffic Descriptor
- Operational Status
- Transmit QoS Class
- Receive QoS Class

4.4.6.1 Interface Index (R)

The Interface Index is the same as that for the Physical interface as defined in section 4.4.2.1.1. It is implicitly the local ATM UNI, unless the optional mode of explicit identification is supported.

4.4.6.2 VPI (R)

This is the value of the Virtual Path Identifier (VPI) for this VCC.

4.4.6.3 VCI (R)

This is the value of the Virtual Channel Identifier (VCI) for this VCC.

4.4.6.4 Transmit Traffic Descriptor (R)

This is a specification of the conformance definition and associated source traffic descriptor parameter values described in section 3.6.2.5 that are applicable to the transmit side of this interface for this VCC.

4.4.6.5 Recieve Traffic Descriptor (R)

This is a specification of the conformance definition and associated source traffic descriptor parameter values described in section 3.6.2.5 that are applicable to the receive side of this interface for this VCC.

4.4.6.6 Operational Status (R)

This object represents the state of the VCC as known by the local device. If the end-to-end status is known then a value of end2endUp(2) or end2endDown(3) is used. If only the local status is known then a value of localUpEnd2endUnknown(4) or localDown(5) is used.

4.4.6.7 Transmit (QoS) Class (R)

This is the QoS Class defined in section 4 of Appendix A, for the transmit side of this interface for thisVCC.

4.4.6.8 Receive (QoS) Class (R)

This is the QoS Class defined in section 4 of Appendix A, for the receive side of this interface for thisVCC.

4.4.7 ILMI Traps (R)

Two traps have been defined for the ILMI in order to indicate a newly configured or deleted permanent VPC or permanent VCC.

For the VPC, the ILMI trap will provide the Virtual Path Identifier (VPI) value of the new or deleted configured VPC at a UNI. For the ILMI trap related to VCC, the trap will also provide the Virtual Channel Identifier (VCI) and the VPI values of the new or deleted configured VCC at a UNI.

4.5 Relationship to Other MIBs

It is required that an ATM device which implements the ILMI MIB will also implement the 'system' group defined in MIB-II.

4.5.1 Relationship to the 'system' group (R)

In MIB-II, the 'system' group is defined as being mandatory for all systems such that each managed entity contains one instance of each object in the 'system' group. Thus, those objects apply to the entity even if the entity's sole functionality is the forwarding of ATM cells.

RFC 1213 is the authoritative source for the definition of objects in the 'system' group. The group consists of the following 7 objects:

(For each textual object for which the device is not configured with a value, the object's value is a string of length 0.)

sysDescr "A textual description of the entity. This value should include the full name and version identification of the system's hardware type, software operating-system, and networking software. It is mandatory that this only contain printable ASCII characters."

sysObjectID "The vendor's authoritative identification of the network management subsystem contained in the entity. This value is allocated within the SMI enterprises subtree (1.3.6.1.4.1) and provides an easy and unambiguous means for determining 'what kind of box' is being managed. For example, if vendor 'Flintstones, Inc.' was assigned the subtree 1.3.6.1.4.1.4242, it could assign the identifier 1.3.6.1.4.1.4242.1.1 to its 'Fred Router'."

sysUpTime	"The time (in hundredths of a second) since the UME was last re-initialized."
sysContact	"The textual identification of the contact person for this managed node, together with information on how to contact this person."
sysName	"An administratively-assigned textual name for this managed node."
sysLocation	"A textual description of the physical location of this device (e.g., `telephone closet, 3rd floor')."
sysServices	"A value which indicates the set of services that this entity primarily offers. The value is a sum of individual values, each representing a particular switching function. The layers are:

layer	**functionality**
1	physical (e.g., repeaters)
2	datalink/subnetwork (e.g., bridges)
3	internet (e.g., IP routers)
4	end-to-end (e.g., IP hosts)
7	applications (e.g., mail relays)."

4.6 Actual MIB

4.6.1 Objects

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the subset of Abstract Syntax Notation One (ASN.1) [23] defined by the Structure of Management Information (SMI) [20]. In particular, each object has a name, a syntax, and an encoding. The name is an object identifier, an administratively assigned name, which specifies an object type. The object type together with an object instance serves to uniquely identify a specific instantiation of the object. For human convenience, we often use a textual string, termed the OBJECT DESCRIPTOR, to also refer to the object type.

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

The encoding of an object type is simply how that object type is represented using the object type's syntax. Implicitly tied to the notion of an object type's syntax and encoding is how the object type is represented when being transmitted on the network.

The SMI specifies the use of the basic encoding rules of ASN.1 [24], subject to the additional requirements imposed by the SNMP [21].

4.6.1.1 Textual Conventions

Several data types are used as textual conventions in this document. These textual conventions have no effect on either the syntax nor the semantics of any managed object. Objects defined using these conventions are always encoded by means of the rules that define their primitive type. Hence, no changes to the SMI or the SNMP are necessary to accommodate these textual conventions which are adopted merely for the convenience of readers.

4.6.1.2 Use of Counters

This MIB defines all counter values using the standard SMI data type: Counter. This data type is defined [20] as representing a non-negative integer which monotonically increases until it reaches a maximum value of 2^32-1 (4294967295 decimal), when it wraps around and starts increasing again from zero.

This definition disallows the clearing of Counter values, in order to prevent interference which would otherwise occur if two network managers were accessing the counters concurrently. Instead, interval values are obtained as the delta between a Counter's values at the beginning and end of a period.

4.6.1.3 Meaning of Transmit/Receive

The terms 'transmit' and 'receive' are used in the following MIB's object descriptors and textual descriptions. In each case, these terms are used from the perspective of the device local to the management information being defined.

4.6.2 Definitions

```
ATM-FORUM-MIB DEFINITIONS ::= BEGIN

IMPORTS
        enterprises, Counter     FROM RFC1155-SMI
        DisplayString            FROM RFC1213-MIB
        OBJECT-TYPE              FROM RFC-1212;

atmForum         OBJECT IDENTIFIER ::= { enterprises 353 }
                 -- a subtree for defining administrative
                 -- object identifiers

atmForumAdmin    OBJECT IDENTIFIER ::= { atmForum 1 }
                 -- a subtree for defining UNI MIB object types

atmForumUni      OBJECT IDENTIFIER ::= { atmForum 2 }

-- Textual Conventions

-- All representations of ATM addresses in this MIB Module use
-- the data type:
```

```
AtmAddress ::= OCTET STRING (SIZE (0 .. 32))

-- Note this data type is used only by the deprecated object
-- atmfPortAddress. Another definition (a refined one) is
-- specified in the separate MIB for Address Registration.

-- Object Identifier definitions

-- The following values are defined for use as possible values
-- of the atmfPortTransmissionType object.

atmfTransmissionTypes OBJECT IDENTIFIER ::= { atmForumAdmin 2 }
        -- unknown transmission type
atmfUnknownType
                OBJECT IDENTIFIER ::= { atmfTransmissionTypes 1}
        -- Sonet STS-3c physical layer at 155.52 Mbps
atmfSonetSTS3c
                OBJECT IDENTIFIER ::= { atmfTransmissionTypes 2 }
        -- DS3 physical layer at 44.736 Mbps
atmfDs3
                OBJECT IDENTIFIER ::= { atmfTransmissionTypes 3 }
        -- 4B/5B encoding physical layer at 100 Mbps
atmf4B5B
                OBJECT IDENTIFIER ::= { atmfTransmissionTypes 4 }
        -- 8B/10B encoding physical layer at 155.52 Mbps
atmf8B10B
                OBJECT IDENTIFIER ::= { atmfTransmissionTypes 5 }

-- The following values are defined for use as possible values
-- of the atmfPortMediaType object.

atmfMediaTypes OBJECT IDENTIFIER ::= { atmForumAdmin 3 }
        -- unknown media type
atmfMediaUnknownType
                OBJECT IDENTIFIER ::= { atmfMediaTypes 1 }
        -- Coaxial cable
atmfMediaCoaxCable
                OBJECT IDENTIFIER ::= { atmfMediaTypes 2 }
        -- Single Mode fiber
atmfMediaSingleMode
                OBJECT IDENTIFIER ::= { atmfMediaTypes 3 }
        -- Multi Mode fiber
atmfMediaMultiMode
                OBJECT IDENTIFIER ::= { atmfMediaTypes 4 }
        -- Shielded Twisted Pair
```

```
atmfMediaStp
                    OBJECT IDENTIFIER ::= { atmfMediaTypes 5 }
          -- Unshielded Twisted Pair
atmfMediaUtp
                    OBJECT IDENTIFIER ::= { atmfMediaTypes 6 }

-- The following values are defined for use as possible values
-- of the atmfVpcTransmitTrafficDescriptorType,
-- atmfVpcReceiveTrafficDescriptorType,
-- atmfVccTransmitTrafficDescriptorType and
-- atmfVccReceiveTrafficDescriptorType objects.

atmfTrafficDescrTypes OBJECT IDENTIFIER ::= { atmForumAdmin 4 }

          -- The "None" Traffic Descriptor Type
atmfNoDescriptor
                    OBJECT IDENTIFIER ::= { atmfTrafficDescrTypes 1 }
--
atmfPeakRate
          OBJECT IDENTIFIER ::= { atmfTrafficDescrTypes 2 }
-- This type is no longer used

--
          -- The No CLP/No SCR Type
atmfNoClpNoScr
                    OBJECT IDENTIFIER ::= { atmfTrafficDescrTypes 3 }
-- The use of the parameter vector for this type:
          -- Parameter #1 - peak cell rate in cells/second for CLP=0+1 traffic
          -- Parameters #2, #3, #4 and #5 are unused

--
          -- The CLP without Tagging/No SCR Type
atmfClpNoTaggingNoScr
                    OBJECT IDENTIFIER ::= { atmfTrafficDescrTypes 4 }
-- The use of the parameter vector for this type:
          -- Parameter #1 - peak cell rate in cells/second for CLP=0+1 traffic
          -- Parameter #2 - peak cell rate in cells/second for CLP=0 traffic
          -- Parameters #3, #4 and #5 are unused

--
          -- The CLP with Tagging/No SCR Type
atmfClpTaggingNoScr
                    OBJECT IDENTIFIER ::= { atmfTrafficDescrTypes 5 }
-- The use of the parameter vector for this type:
          -- Parameter #1 - peak cell rate in cells/second for CLP=0+1 traffic
```

```
        -- Parameter #2 - peak cell rate in cells/second for
        -- CLP=0 traffic, excess tagged as CLP=1
        -- Parameters #3, #4 and #5 are unused

--
        -- The SCR/No CLP Type
atmfNoClpScr
                OBJECT IDENTIFIER ::= { atmfTrafficDescrTypes 6 }
-- The use of the parameter vector for this type:
        -- Parameter #1 - peak cell rate in cells/second for CLP=0+1 traffic
        -- Parameter #2 - sustainable cell rate in cells/second for CLP=0+1 traffic
        -- Parameter #3 - maximum burst size in cells
        -- Parameters #4 and #5 are unused

--
        -- The CLP without Tagging/SCR Type
atmfClpNoTaggingScr
                OBJECT IDENTIFIER ::= { atmfTrafficDescrTypes 7 }
-- The use of the parameter vector for this type:
        -- Parameter #1 - peak cell rate in cells/second for CLP=0+1 traffic
        -- Parameter #2 - sustainable cell rate in cells/second for CLP=0 traffic
        -- Parameter #3 - maximum burst size in cells
        -- Parameters #4 and #5 are unused

--
        -- The CLP with Tagging/SCR Type
atmfClpTaggingScr
                OBJECT IDENTIFIER ::= { atmfTrafficDescrTypes 8 }
-- The use of the parameter vector for this type:
        -- Parameter #1 - peak cell rate in cells/second for CLP=0+1 traffic
        -- Parameter #2 - sustainable cell rate in cells/second for CLP=0
        --      traffic, excess tagged as CLP=1
        -- Parameter #3 - maximum burst size in cells
        -- Parameters #4 and #5 are unused

--                      The MIB groups
atmfPhysicalGroup       OBJECT IDENTIFIER ::= { atmForumUni 1 }
atmfAtmLayerGroup       OBJECT IDENTIFIER ::= { atmForumUni 2 }
atmfAtmStatsGroup       OBJECT IDENTIFIER ::= { atmForumUni 3 }
atmfVpcGroup            OBJECT IDENTIFIER ::= { atmForumUni 4 }
atmfVccGroup            OBJECT IDENTIFIER ::= { atmForumUni 5 }
```

```
--                          The Physical Port Group
-- This group is mandatory for all UNI devices.
--
-- The Physical Port Table

atmfPortTable OBJECT-TYPE
        SYNTAX                  SEQUENCE OF AtmfPortEntry
        ACCESS                  not-accessible
        STATUS                  mandatory
        DESCRIPTION
                "A table of physical layer status and parameter information for the UNI's
                physical interface."
        ::= { atmfPhysicalGroup 1 }

atmfPortEntry OBJECT-TYPE
        SYNTAX                  AtmfPortEntry
        ACCESS                  not-accessible
        STATUS                  mandatory
        DESCRIPTION
                "An entry in the table, containing information about the physical layer of a
                UNI interface."
        INDEX { atmfPortIndex }
        ::= { atmfPortTable 1 }

AtmfPortEntry ::=
        SEQUENCE {
                atmfPortIndex
                        INTEGER,
                atmfPortAddress
                        AtmAddress,
                atmfPortTransmissionType
                        OBJECT IDENTIFIER,
                atmfPortMediaType
                        OBJECT IDENTIFIER,
                atmfPortOperStatus
                        INTEGER,
                atmfPortSpecific
                        OBJECT IDENTIFIER
                atmfPortMyIfName
                        DISPLAYSTRING
        }

atmfPortIndex OBJECT-TYPE
        SYNTAX                  INTEGER (0..2147483647)
        ACCESS                  read-only
        STATUS                  mandatory
        DESCRIPTION
```

```
                    "A unique value which identifies this port. The value of 0 has the special
                    meaning of identifying the local UNI."
          ::= { atmfPortEntry 1 }

atmfPortAddress OBJECT-TYPE
          SYNTAX              AtmAddress
          ACCESS              read-only
          STATUS              deprecated
          DESCRIPTION
                    "This object should not be implemented except as required for backward
                    compatibility with version 2.0 of the UNI specification. The Address
                    Group, defined as part of the separate Address Registration MIB should be
                    used instead."

          ::= { atmfPortEntry 2 }

atmfPortTransmissionType OBJECT-TYPE
          SYNTAX              OBJECT IDENTIFIER
          ACCESS              read-only
          STATUS              mandatory
          DESCRIPTION
                    "The transmission type of this port. For example, for a port using the Sonet
                    STS-3c physical layer at 155.52 Mbs, this object would have the Object
                    Identifier value: atmfSonetSTS3c."
          ::= { atmfPortEntry 3 }

atmfPortMediaType OBJECT-TYPE
          SYNTAX              OBJECT IDENTIFIER
          ACCESS              read-only
          STATUS              mandatory
          DESCRIPTION
                    "The type of media being used on this port. For example, for a port using
                    coaxial cable, this object would have the Object Identifier value:
                    atmfMediaCoaxCable."
          ::= { atmfPortEntry 4 }

atmfPortOperStatus OBJECT-TYPE
          SYNTAX              INTEGER {
                                        other(1),
                                        inService(2),
                                        outOfService(3),
                                        loopBack(4)
                              }
          ACCESS              read-only
```

```
        STATUS              mandatory
        DESCRIPTION
                "The operational (i.e., actual) state of this port.

                The ILMI should not alarm on a physical interface for when the value of
                this object is outOfService(3). This capability is useful if equipment is to be
                disconnected, or for troubleshooting purposes.

                A value of loopBack(4) indicates that a local loopback is in place. "
        ::= { atmfPortEntry 5 }

atmfPortSpecific OBJECT-TYPE
        SYNTAX   OBJECT IDENTIFIER
        ACCESS   read-only
        STATUS   mandatory
        DESCRIPTION
                "This object 'points' to additional transmission and/or media specific
                information relating to this port. In particular, this object's value is the name
                of a specific instance of the first columnar object of a MIB table with such
                additional information, where the specific instance is the one which corre-
                sponds to this port.

                For example, for a DS3 interface, this object would contain the value, as
                defined in RFC 1407:
                        dsx3LineIndex.i
                where i would be the integer value uniquely identifying the DS3 interface
                corresponding to this port. If no additional transmission and/or media
                specific information is available, this object has the value { 0 0 }."
        ::= { atmfPortEntry 6 }


atmfPortMyIfName  OBJECT-TYPE
        SYNTAX        DisplayString
        ACCESS        read-only
        STATUS        mandatory
        DESCRIPTION
                "A textual name of this interface. If this 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. This interface must be uniquely named on this system
                to distinguish parallel links with a neighboring system. If this interface does
                not have a textual name, the value of this object is a zero length string."
        ::= { atmfPortEntry 7 }
 -- Note: Typical UME will support only one of the following two objects
```

```
atmfMyIpNmAddress  OBJECT-TYPE
        SYNTAX          IpAddress
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION
                "An IP Address to which a Network Management Station can send Network
                Management protocol, e.g. SNMP messages  to UDP port 161, in order to
                access network management information concerning the operation of the
                ATM device local  to this UME."
        ::= { atmfPhysicalGroup 2 }

atmfMyOsiNmNsapAddress  OBJECT-TYPE
        SYNTAX          NsapAddress
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION
                "An NSAP Address to which a Network Management Station can send
                Network Management protocol messages in order to access network
                management information concerning the operation of the ATM device local
                to this UME."
        ::= { atmfPhysicalGroup 3 }

--                      The ATM Layer Group
-- This group is mandatory for all UNI devices.
--
-- ATM-layer specific information for the UNI interface
atmfAtmLayerTable OBJECT-TYPE
        SYNTAX                  SEQUENCE OF AtmfAtmLayerEntry
        ACCESS                  not-accessible
        STATUS                  mandatory
        DESCRIPTION
                "A table of ATM layer status and parameter information for the UNI's
                physical interface."
        ::= { atmfAtmLayerGroup 1 }

atmfAtmLayerEntry OBJECT-TYPE
        SYNTAX                  AtmfAtmLayerEntry
        ACCESS                  not-accessible
        STATUS                  mandatory
        DESCRIPTION
                "An entry in the table, containing information about the ATM layer of a
                UNI interface."
        INDEX { atmfAtmLayerIndex }
        ::= { atmfAtmLayerTable 1 }
```

```
AtmfAtmLayerEntry ::=
        SEQUENCE {
                atmfAtmLayerIndex
                        INTEGER,
                atmfAtmLayerMaxVPCs
                        INTEGER,
                atmfAtmLayerMaxVCCs
                        INTEGER,
                atmfAtmLayerConfiguredVPCs
                        INTEGER,
                atmfAtmLayerConfiguredVCCs
                        INTEGER,
                atmfAtmLayerMaxVpiBits
                        INTEGER,
                atmfAtmLayerMaxVciBits
                        INTEGER,
                atmfAtmLayerUniType
                        INTEGER
                atmfAtmLayerUniVersion
                        INTEGER
        }
atmfAtmLayerIndex OBJECT-TYPE
        SYNTAX              INTEGER (0..2147483647)
        ACCESS              read-only
        STATUS              mandatory
        DESCRIPTION
                "The unique value which identifies the UNI port. The value of 0 has the
                special meaning of identifying the local UNI."
        ::= { atmfAtmLayerEntry 1 }

atmfAtmLayerMaxVPCs OBJECT-TYPE
        SYNTAX              INTEGER (0..255)
        ACCESS              read-only
        STATUS              mandatory
        DESCRIPTION
                "The maximum number of switched and permanent VPCs supported on this
                UNI."
        ::= { atmfAtmLayerEntry 2 }

atmfAtmLayerMaxVCCs OBJECT-TYPE
        SYNTAX              INTEGER (0..16777215)
        ACCESS              read-only
        STATUS              mandatory
```

```
        DESCRIPTION
                "The maximum number of switched and permanent VCCs supported on
                this UNI."
        ::= { atmfAtmLayerEntry 3 }

atmfAtmLayerConfiguredVPCs OBJECT-TYPE
        SYNTAX          INTEGER (0..255)
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION
                "The number of VPCs configured for use on this UNI."
        ::= { atmfAtmLayerEntry 4 }

atmfAtmLayerConfiguredVCCs OBJECT-TYPE
        SYNTAX          INTEGER (0..16777215)
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION
                "The number of permanent VCCs configured for use on this UNI."
        ::= { atmfAtmLayerEntry 5 }

atmfAtmLayerMaxVpiBits OBJECT-TYPE
        SYNTAX          INTEGER (0..8)
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION
                "The number of active permanent VPI bits on this interface."
        ::= {atmfAtmLayerEntry 6 }

atmfAtmLayerMaxVciBits OBJECT-TYPE
        SYNTAX          INTEGER (0..16)
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION
                "The number of active VCI bits on this interface."
        ::= {atmfAtmLayerEntry 7 }

atmfAtmLayerUniType OBJECT-TYPE
        SYNTAX          INTEGER {public(1), private(2)}
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION
                "The type of the ATM UNI, either public or private."
        ::= { atmfAtmLayerEntry 8 }
```

```
atmfAtmLayerUniVersion  OBJECT-TYPE
        SYNTAX                 INTEGER {version2point0(1), version3point0(2),
                               version3point1(3)  }
        ACCESS                 read-only
        STATUS                 mandatory
        DESCRIPTION
                "An indication of the latest version of the ATM UNI Specification that is
                supported on this UNI.  If this value is not present, a version of the UNI
                earlier than 3.1 is supported.  If a value greater than version3point1 is
                present, then UNI 3.1 communication should be attempted.

                If the peer UME's value of this object is the same as, or later than the local
                UME's value, then the version corresponding to the local UME's value
                should be attempted.  Otherwise, if the peer UME's value of this object is
                earlier, and supported locally, then the local UME should attempt the
                version corresponding to the peer UME's value.  Otherwise, compatibility of
                the two UMEs cannot be assumed."
        ::= { atmfAtmLayerEntry 9 }

--                              The ATM Statistics Group
-- This group is optional. However, if any objects in this group
-- are supported, then all objects in the group must be supported.
--
-- ATM-layer statistics for the UNI interface

atmfAtmStatsTable OBJECT-TYPE
        SYNTAX                 SEQUENCE OF AtmfAtmStatsEntry
        ACCESS                 not-accessible
        STATUS                 mandatory
        DESCRIPTION
                "A table of ATM layer statistics information for
                the UNI's physical interface."
        ::= { atmfAtmStatsGroup 1 }

atmfAtmStatsEntry OBJECT-TYPE
        SYNTAX                 AtmfAtmStatsEntry
        ACCESS                 not-accessible
        STATUS                 mandatory
        DESCRIPTION
                "An entry in the table, containing statistics for the ATM layer of a UNI
                interface."
        INDEX { atmfAtmStatsIndex }
        ::= { atmfAtmStatsTable 1 }
```

```
AtmfAtmStatsEntry ::=
        SEQUENCE {
                atmfAtmStatsIndex
                        INTEGER,
                atmfAtmStatsReceivedCells
                        Counter,
                atmfAtmStatsDroppedReceivedCells
                        Counter,
                atmfAtmStatsTransmittedCells
                        Counter
        }
atmfAtmStatsIndex OBJECT-TYPE
        SYNTAX          INTEGER (0..2147483647)
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION
                "The unique value which identifies the UNI port. The value of 0 has the
                special meaning of identifying the local UNI."
        ::= { atmfAtmStatsEntry 1 }

atmfAtmStatsReceivedCells OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION
                "The accumulated number of ATM cells received on this UNI which were
                assigned and not dropped."
        ::= { atmfAtmStatsEntry 2 }

atmfAtmStatsDroppedReceivedCells OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION
                "The accumulated number of ATM cells which were dropped for the
                reasons defined in section 4.4.4.2."
        ::= { atmfAtmStatsEntry 3 }

atmfAtmStatsTransmittedCells OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION
                "The accumulated number of assigned ATM cells which were transmitted
                across this interface."
```

```
        ::= { atmfAtmStatsEntry 4 }


--                        The Virtual Path Group
-- This group is mandatory for all UNI devices.
--
-- Information concerning Virtual Path Connections

atmfVpcTable OBJECT-TYPE
        SYNTAX          SEQUENCE OF AtmfVpcEntry
        ACCESS          not-accessible
        STATUS          mandatory
        DESCRIPTION
                "A table of status and parameter information on the virtual path connections
                which cross this UNI. There is one entry in this table for each permanent
                virtual path connection."
        ::= { atmfVpcGroup 1 }

atmfVpcEntry OBJECT-TYPE
        SYNTAX          AtmfVpcEntry
        ACCESS          not-accessible
        STATUS          mandatory
        DESCRIPTION
                "An entry in the table, containing information about a particular virtual path
                connection."
        INDEX { atmfVpcPortIndex, atmfVpcVpi }
        ::= { atmfVpcTable 1 }

AtmfVpcEntry ::=
        SEQUENCE {
                atmfVpcPortIndex
                        INTEGER,
                atmfVpcVpi
                        INTEGER,
                atmfVpcOperStatus
                        INTEGER,
                atmfVpcTransmitTrafficDescriptorType
                        OBJECT IDENTIFIER,
                atmfVpcTransmitTrafficDescriptorParam1
                        INTEGER,
                atmfVpcTransmitTrafficDescriptorParam2
                        INTEGER,
                atmfVpcTransmitTrafficDescriptorParam3
                        INTEGER,
```

```
                atmfVpcTransmitTrafficDescriptorParam4
                        INTEGER,
                atmfVpcTransmitTrafficDescriptorParam5
                        INTEGER,
                atmfVpcReceiveTrafficDescriptorType
                        OBJECT IDENTIFIER,
                atmfVpcReceiveTrafficDescriptorParam1
                        INTEGER,
                atmfVpcReceiveTrafficDescriptorParam2
                        INTEGER,
                atmfVpcReceiveTrafficDescriptorParam3
                        INTEGER,
                atmfVpcReceiveTrafficDescriptorParam4
                        INTEGER,
                atmfVpcReceiveTrafficDescriptorParam5
                        INTEGER,
                atmfVpcQoSCategory
                        INTEGER,
                atmfVpcTransmitQoSClass
                        INTEGER,
                atmfVpcReceiveQoSClass
                        INTEGER
        }

atmfVpcPortIndex OBJECT-TYPE
        SYNTAX                  INTEGER (0..2147483647)
        ACCESS                  read-only
        STATUS                  mandatory
        DESCRIPTION
                "The unique value which identifies the UNI port. The value of 0 has the
                special meaning of identifying the local UNI."
        ::= { atmfVpcEntry 1 }

atmfVpcVpi OBJECT-TYPE
        SYNTAX                  INTEGER (0..255)
        ACCESS                  read-only
        STATUS                  mandatory
        DESCRIPTION
                "The VPI value of this Virtual Path Connection at the local UNI."
        ::= { atmfVpcEntry 2 }

atmfVpcOperStatus OBJECT-TYPE
        SYNTAX INTEGER {
                                unknown(1),
                                end2endUp(2),
```

```
                            end2endDown(3),
                            localUpEnd2endUnknown(4),
                            localDown(5)
                        }
    ACCESS              read-only
    STATUS              mandatory
    DESCRIPTION
        "The present actual operational status of the VPC.

        A value of end2endUp(2) or end2endDown(3) would be used if the end-to-
        end status is known. If only local status information is available, a value of
        localUpEnd2endUnknown(4) or localDown(5) would be used."

        ::= { atmfVpcEntry 3 }

atmfVpcTransmitTrafficDescriptorType OBJECT-TYPE
    SYNTAX              OBJECT IDENTIFIER
    ACCESS              read-only
    STATUS              mandatory
    DESCRIPTION
        "The type of traffic management, applicable to the transmit direction of this
        VPC. The type may indicate none, or a type with one or more parameters.
        These parameters are specified as a parameter vector, in the corresponding
        instances of the objects:
        atmfVpcTransmitTrafficDescriptorParam1,
        atmfVpcTransmitTrafficDescriptorParam2,
        atmfVpcTransmitTrafficDescriptorParam3,
        atmfVpcTransmitTrafficDescriptorParam4, and
        atmfVpcTransmitTrafficDescriptorParam5."
    ::= { atmfVpcEntry 4 }

atmfVpcTransmitTrafficDescriptorParam1 OBJECT-TYPE
    SYNTAX              INTEGER (0..2147483647)
    ACCESS              read-only
    STATUS              mandatory
    DESCRIPTION
        "The first parameter of the transmit parameter vector for this VPC, used
        according to the value of atmfVpcTransmitTrafficDescriptorType."
    ::= { atmfVpcEntry 5 }

atmfVpcTransmitTrafficDescriptorParam2 OBJECT-TYPE
    SYNTAX              INTEGER (0..2147483647)
    ACCESS              read-only
    STATUS              mandatory
```

```
        DESCRIPTION
                "The second parameter of the transmit parameter vector for this VPC, used
                according to the value of atmfVpcTransmitTrafficDescriptorType."
        ::= { atmfVpcEntry 6 }

atmfVpcTransmitTrafficDescriptorParam3 OBJECT-TYPE
        SYNTAX          INTEGER (0..2147483647)
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION
                "The third parameter of the transmit parameter vector for this VPC, used
                according to the value of atmfVpcTransmitTrafficDescriptorType."
        ::= { atmfVpcEntry 7 }

atmfVpcTransmitTrafficDescriptorParam4 OBJECT-TYPE
        SYNTAX          INTEGER (0..2147483647)
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION
                "The fourth parameter of the transmit parameter vector for this VPC, used
                according to the value of atmfVpcTransmitTrafficDescriptorType."
        ::= { atmfVpcEntry 8 }

atmfVpcTransmitTrafficDescriptorParam5 OBJECT-TYPE
        SYNTAX          INTEGER (0..2147483647)
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION
                "The fifth parameter of the transmit parameter vector for this VPC, used
                according to the value of atmfVpcTransmitTrafficDescriptorType."
        ::= { atmfVpcEntry 9 }

atmfVpcReceiveTrafficDescriptorType OBJECT-TYPE
        SYNTAX          OBJECT IDENTIFIER
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION
                "The type of traffic management, applicable to the traffic in the receive
                direction of this VPC. The type may indicate none, or a type with one or
                more parameters. These parameters are specified as a parameter vector, in
                the corresponding instances of the objects:
                atmfVpcReceiveTrafficDescriptorParam1,
                atmfVpcReceiveTrafficDescriptorParam2,
                atmfVpcReceiveTrafficDescriptorParam3,
```

```
                atmfVpcReceiveTrafficDescriptorParam4, and
                atmfVpcReceiveTrafficDescriptorParam5."
        ::= { atmfVpcEntry 10 }

atmfVpcReceiveTrafficDescriptorParam1 OBJECT-TYPE
        SYNTAX              INTEGER (0..2147483647)
        ACCESS              read-only
        STATUS              mandatory
        DESCRIPTION
                "The first parameter of the receive parameter vector for this VPC, used
                according to the value of atmfVpcReceiveTrafficDescriptorType."
        ::= { atmfVpcEntry 11 }

atmfVpcReceiveTrafficDescriptorParam2 OBJECT-TYPE
        SYNTAX              INTEGER (0..2147483647)
        ACCESS              read-only
        STATUS              mandatory
        DESCRIPTION
                "The second parameter of the receive parameter vector for this VPC, used
                according to the value of atmfVpcReceiveTrafficDescriptorType."
        ::= { atmfVpcEntry 12 }

atmfVpcReceiveTrafficDescriptorParam3 OBJECT-TYPE
        SYNTAX              INTEGER (0..2147483647)
        ACCESS              read-only
        STATUS              mandatory
        DESCRIPTION
                "The third parameter of the receive parameter vector for this VPC, used
                according to the value of atmfVpcReceiveTrafficDescriptorType."
        ::= { atmfVpcEntry 13 }

atmfVpcReceiveTrafficDescriptorParam4 OBJECT-TYPE
        SYNTAX              INTEGER (0..2147483647)
        ACCESS              read-only
        STATUS              mandatory
        DESCRIPTION
                "The fourth parameter of the receive parameter vector for this VPC, used
                according to the value of atmfVpcReceiveTrafficDescriptorType."
        ::= { atmfVpcEntry 14 }

atmfVpcReceiveTrafficDescriptorParam5 OBJECT-TYPE
        SYNTAX              INTEGER (0..2147483647)
        ACCESS              read-only
        STATUS              mandatory
```

```
        DESCRIPTION
                "The fifth parameter of the receive parameter vector for this VPC, used
                according to the value of atmfVpcReceiveTrafficDescriptorType."
        ::= { atmfVpcEntry 15 }

atmfVpcQoSCategory OBJECT-TYPE
        SYNTAX          INTEGER {
                                other(1),
                                deterministic (2),
                                statistical (3),
                                unspecified (4)
                        }
        ACCESS          read-only
        STATUS          deprecated
        DESCRIPTION
                "This object should not be implemented except as required for backward
                compatibility with version 2.0 of the UNI specification."
        ::= { atmfVpcEntry 16 }

atmfVpcTransmitQoSClass OBJECT-TYPE
        SYNTAX          INTEGER (0..255)
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION
                "The QoS Class, as defined in section 4 of Appendix A, for the transmit
                direction of this VPC connection at the local UNI."
        ::= { atmfVpcEntry 17 }

atmfVpcReceiveQoSClass OBJECT-TYPE
        SYNTAX          INTEGER (0..255)
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION
                "The QoS Class, as defined in section 4 of Appendix A, for the receive
                direction of this VPC connection at the local UNI."
        ::= { atmfVpcEntry 18 }

--                      The Virtual Channel Group
-- This group is mandatory for all UNI devices.
--
-- Information concerning Virtual Channel Connections
atmfVccTable OBJECT-TYPE
        SYNTAX          SEQUENCE OF AtmfVccEntry
        ACCESS          not-accessible
        STATUS          mandatory
```

```
        DESCRIPTION
                "A table of status and parameter information on the virtual channel connec-
                tions which are visible at this UNI. There is one entry in this table for each
                permanent virtual channel connection, including reserved VCCs that are
                supported;   e.g., signalling, OAM flows, and ILMI, but not unassigned
                cells."
        ::= { atmfVccGroup 1 }

atmfVccEntry OBJECT-TYPE
        SYNTAX                AtmfVccEntry
        ACCESS                not-accessible
        STATUS                mandatory
        DESCRIPTION
                "An entry in the table, containing information about a particular virtual
                channel connection."
        INDEX { atmfVccPortIndex, atmfVccVpi, atmfVccVci }
        ::= { atmfVccTable 1 }

AtmfVccEntry ::=
        SEQUENCE {
                atmfVccPortIndex
                        INTEGER,
                atmfVccVpi
                        INTEGER,
                atmfVccVci
                        INTEGER,
                atmfVccOperStatus
                        INTEGER,
                atmfVccTransmitTrafficDescriptorType
                        OBJECT IDENTIFIER,
                atmfVccTransmitTrafficDescriptorParam1
                        INTEGER,
                atmfVccTransmitTrafficDescriptorParam2
                        INTEGER,
                atmfVccTransmitTrafficDescriptorParam3
                        INTEGER,
                atmfVccTransmitTrafficDescriptorParam4
                        INTEGER,
                atmfVccTransmitTrafficDescriptorParam5
                        INTEGER,
                atmfVccReceiveTrafficDescriptorType
                        OBJECT IDENTIFIER,
                atmfVccReceiveTrafficDescriptorParam1
                        INTEGER,
```

```
                atmfVccReceiveTrafficDescriptorParam2
                        INTEGER,
                atmfVccReceiveTrafficDescriptorParam3
                        INTEGER,
                atmfVccReceiveTrafficDescriptorParam4
                        INTEGER,
                atmfVccReceiveTrafficDescriptorParam5
                        INTEGER,
                atmfVccQoSCategory
                        INTEGER,
                atmfVccTransmitQoSClass
                        INTEGER,
                atmfVccReceiveQoSClass
                        INTEGER
        }

atmfVccPortIndex OBJECT-TYPE
        SYNTAX                  INTEGER (0..2147483647)
        ACCESS                  read-only
        STATUS                  mandatory
        DESCRIPTION
                "The unique value which identifies the UNI port. The value of 0 has the
                special meaning of identifying the local UNI."
        ::= { atmfVccEntry 1 }

atmfVccVpi OBJECT-TYPE
        SYNTAX                  INTEGER (0..255)
        ACCESS                  read-only
        STATUS                  mandatory
        DESCRIPTION
                "The VPI value of this Virtual Channel Connection at the local UNI."
        ::= { atmfVccEntry 2 }

atmfVccVci OBJECT-TYPE
        SYNTAX                  INTEGER (0..65535)
        ACCESS                  read-only
        STATUS                  mandatory
        DESCRIPTION
                "The VCI value of this Virtual Channel Connection at the local UNI."
        ::= { atmfVccEntry 3 }

atmfVccOperStatus OBJECT-TYPE
        SYNTAX INTEGER {
                        unknown(1),
                        end2endUp(2),
```

```
                    end2endDown(3),
                    localUpEnd2endUnknown(4),
                    localDown(5)
                    }
        ACCESS      read-only
        STATUS      mandatory
        DESCRIPTION
                "The present actual operational status of the VCC. A value of
                end2endUp(2) or end2endUp(3) is used if the end to end status is known.

                If only local status is known a value of localUpEnd2endUnknown(4) or
                localDown(5) is used.

        ::= { atmfVccEntry 4 }

atmfVccTransmitTrafficDescriptorType OBJECT-TYPE
        SYNTAX              OBJECT IDENTIFIER
        ACCESS              read-only
        STATUS              mandatory
        DESCRIPTION
                "The type of traffic management, applicable to the transmit direction of this
                VCC. The type may indicate none, or a type with one or more parameters.
                These parameters are specified as a parameter vector, in the corresponding
                instances of the objects:
                atmfVccTransmitTrafficDescriptorParam1,
                atmfVccTransmitTrafficDescriptorParam2,
                atmfVccTransmitTrafficDescriptorParam3,
                atmfVccTransmitTrafficDescriptorParam4, and
                atmfVccTransmitTrafficDescriptorParam5."
        ::= { atmfVccEntry 5 }

atmfVccTransmitTrafficDescriptorParam1 OBJECT-TYPE
        SYNTAX              INTEGER (0..2147483647)
        ACCESS              read-only
        STATUS              mandatory
        DESCRIPTION
                "The first parameter of the transmit parameter vector for this VCC, used
                according to the value of atmfVccTransmitTrafficDescriptorType."
        ::= { atmfVccEntry 6 }

atmfVccTransmitTrafficDescriptorParam2 OBJECT-TYPE
        SYNTAX              INTEGER (0..2147483647)
        ACCESS              read-only
        STATUS              mandatory
```

```
        DESCRIPTION
                "The second parameter of the transmit parameter vector for this VCC, used
                according to the value of atmfVccTransmitTrafficDescriptorType."
        ::= { atmfVccEntry 7 }

atmfVccTransmitTrafficDescriptorParam3 OBJECT-TYPE
        SYNTAX              INTEGER (0..2147483647)
        ACCESS              read-only
        STATUS              mandatory
        DESCRIPTION
                "The third parameter of the transmit parameter vector for this VCC, used
                according to the value of atmfVccTransmitTrafficDescriptorType."
        ::= { atmfVccEntry 8 }

atmfVccTransmitTrafficDescriptorParam4 OBJECT-TYPE
        SYNTAX              INTEGER (0..2147483647)
        ACCESS              read-only
        STATUS              mandatory
        DESCRIPTION
                "The fourth parameter of the transmit parameter vector for this VCC, used
                according to the value of atmfVccTransmitTrafficDescriptorType."
        ::= { atmfVccEntry 9 }

atmfVccTransmitTrafficDescriptorParam5 OBJECT-TYPE
        SYNTAX              INTEGER (0..2147483647)
        ACCESS              read-only
        STATUS              mandatory
        DESCRIPTION
                "The fifth parameter of the transmit parameter vector for this VCC, used
                according to the value of atmfVccTransmitTrafficDescriptorType."
        ::= { atmfVccEntry 10 }

atmfVccReceiveTrafficDescriptorType OBJECT-TYPE
        SYNTAX              OBJECT IDENTIFIER
        ACCESS              read-only
        STATUS              mandatory
        DESCRIPTION
                "The type of traffic management, applicable to the traffic in the receive
                direction of this VCC. The type may indicate none, or a type with one or
                more parameters. These parameters are specified as a parameter vector, in
                the corresponding instances of the objects:
                atmfVccReceiveTrafficDescriptorParam1,
                atmfVccReceiveTrafficDescriptorParam2,
                atmfVccReceiveTrafficDescriptorParam3,
                atmfVccReceiveTrafficDescriptorParam4, and
```

```
                atmfVccReceiveTrafficDescriptorParam5."
        ::= { atmfVccEntry 11 }

atmfVccReceiveTrafficDescriptorParam1 OBJECT-TYPE
        SYNTAX                  INTEGER (0..2147483647)
        ACCESS                  read-only
        STATUS                  mandatory
        DESCRIPTION
                "The first parameter of the receive parameter vector for this VCC, used
                according to the value of atmfVccReceiveTrafficDescriptorType."
        ::= { atmfVccEntry 12 }

atmfVccReceiveTrafficDescriptorParam2 OBJECT-TYPE
        SYNTAX                  INTEGER (0..2147483647)
        ACCESS                  read-only
        STATUS                  mandatory
        DESCRIPTION
                "The second parameter of the receive parameter vector for this VCC, used
                according to the value of atmfVccReceiveTrafficDescriptorType."
        ::= { atmfVccEntry 13 }

atmfVccReceiveTrafficDescriptorParam3 OBJECT-TYPE
        SYNTAX                  INTEGER (0..2147483647)
        ACCESS                  read-only
        STATUS                  mandatory
        DESCRIPTION
                "The third parameter of the receive parameter vector for this VCC, used
                according to the value of atmfVccReceiveTrafficDescriptorType."
        ::= { atmfVccEntry 14 }

atmfVccReceiveTrafficDescriptorParam4 OBJECT-TYPE
        SYNTAX                  INTEGER (0..2147483647)
        ACCESS                  read-only
        STATUS                  mandatory
        DESCRIPTION
                "The fourth parameter of the receive parameter vector for this VCC, used
                according to the value of atmfVccReceiveTrafficDescriptorType."
        ::= { atmfVccEntry 15 }

atmfVccReceiveTrafficDescriptorParam5 OBJECT-TYPE
        SYNTAX                  INTEGER (0..2147483647)
        ACCESS                  read-only
        STATUS                  mandatory
```

```
        DESCRIPTION
                "The fifth parameter of the receive parameter vector for this VCC, used
                according to the value of atmfVccReceiveTrafficDescriptorType."
        ::= { atmfVccEntry 16 }

atmfVccQoSCategory OBJECT-TYPE
        SYNTAX              INTEGER {
                                    other(1),
                                    deterministic (2),
                                    statistical (3),
                                    unspecified (4)
                            }
        ACCESS              read-only
        STATUS              deprecated
        DESCRIPTION
                "This object should not be implemented except as required for backward
                compatibility with version 2.0 of the UNI specification."

        ::= { atmfVccEntry 17 }

atmfVccTransmitQoSClass OBJECT-TYPE
        SYNTAX              INTEGER (0..255)
        ACCESS              read-only
        STATUS              mandatory
        DESCRIPTION
                "The QoS Class, as defined in section 4 of Appendix A, for the transmit
                direction of this VCC connection at the local UNI."
        ::= { atmfVccEntry 18 }

atmfVccReceiveQoSClass OBJECT-TYPE
        SYNTAX              INTEGER (0..255)
        ACCESS              read-only
        STATUS              mandatory
        DESCRIPTION
                "The QoS Class, as defined in section 4 of Appendix A, for the receive
                direction of this VCC connection at the local UNI."
        ::= { atmfVccEntry 19 }

atmfVpcChange TRAP-TYPE
        ENTERPRISE          atmForum
        VARIABLES           {atmfVpcPortIndex, atmfVpcVpi, atmfVpcStatus }
        DESCRIPTION
                "An atmfVpcChange trap indicates that a VPC is added or deleted at this
                UNI.  The variables included in the trap identify the VPI value of the new
                or deleted configured VPC at this UNI."
```

```
        ::= 1

atmfVccChange TRAP-TYPE
        ENTERPRISE          atmForum
        VARIABLES           {atmfVccPortIndex, atmfVccVci, atmfVccVpi,
atmfVccStatus }
        DESCRIPTION
                "An atmfVccChange trap indicates that a VCC is added or deleted at this
                UNI.  The variables included in the trap identify the VCI and VPI values of
                the new or deleted configured VCC at this UNI."
        ::= 2

END
```

4.7 ILMI Protocol

The use of SNMP messages will be according to the following requirements.

4.7.1 Use of VCCs

One VCC will be used for sending AAL-encapsulated SNMP messages between adjacent UMEs. This VCC is used for requests, responses, and traps, differentiated according to the SNMP PDU type.

(R) Encapsulation of SNMP ILMI messages in AAL5 as defined in T1S1/92-283, [27], and T1S1/92-285, [28], is required.

(O) Encapsulation of SNMP ILMI messages in the common part of AAL3/AAL4 as defined in I.363 is an option.

(R) At all times one VCC will be provisioned for the ILMI. The default value for provisioning this VCC is VPI=0, VCI=16, however, the VPI/VCI value must be configurable.

(R) The cells carrying ILMI messages shall have cell loss priority (CLP = 0)

(R) The throughput of SNMP traffic on the ILMI VCC should be no.more than approximately one percent of the link bandwidth.

4.7.2 Message Format (R)

The message format specified in RFC 1157 will be used. That is, messages will be formatted according to SNMP version 1, not SNMP version 2. Any use of SMNP version 2 is for future study.

All SNMP messages will use the community name "ILMI", that is, the OCTET STRING value: '494C4D49'h.

In all SNMP Traps, the agent-addr field (which has syntax NetworkAddress), will always have the IpAddress value: 0.0.0.0.

In all SNMP Traps, the time-stamp field in the Trap-PDU will contain the value of the agent's sysUpTime MIB object at the time of trap generation (sysUpTime is defined in the system group of MIB-II).

In any standard SNMP Traps, (e.g., the coldStart Trap), the enterprise field in the Trap-PDU will contain the value of the agent's sysObjectID MIB object (sysObjectID is defined in the system group of MIB-II).

The supported traps are coldStart and enterpriseSpecific.

4.7.3 Message Sizes (R)

All UNI implementations will be able to accept SNMP messages of size up to and including 484 octets. Larger messages should not be sent unless the originator has information (e.g., via some out-of-band mechanism) that the receiver supports larger messages.

4.7.4 ILMI Traffic Requirements

The traffic requirements relating to the ATM connection used for ILMI communication are as follows:

(R) The VCC used for ILMI communication shall support a sustainable cell rate, R(s), no more than 1% of the UNI physical line rate and a peak cell rate, R(p), no more than 5% of the UNI physical line rate.

(R) The ILMI traffic burst length, L(b), shall be no more than 484 octets.

(R) The ILMI traffic bursts inter-arrival time, T(b), should be greater than or equal to (L(b)/(R(p) x 1%)), where L(b) and R(p) are respectively the ILMI burst length and the ILMI traffic peak rate.

4.7.5 Message Response Time

Response time refers to the elapsed time from the submission of an SNMP message (e.g., Get, Get-Next, or Set-Request message) by a UME across a UNI to the receipt of the corresponding SNMP message (e.g., Get-Response message) from the adjacent UME. An SNMP Get, Get-Next, or Set-Request message is defined in this context as a request concerning a single object. The following specifies ILMI response time requirements.

(R) The UME should support maximum Response Times of 1 second for 95% of all SNMP Get, GetNext or SetRequest requests containing a single object received from an adjacent UME independent of the UNI's physical line rate.

4.7.6 Object Value Data Currentness

Data currentness refers to the maximum elapsed time since an object value in the ATM UNI ILMI MIB was known to be current. The following specifies the requirements on the Data Currentness of the ILMI objects and the event notifications.

(R) The ATM UNI ILMI MIB objects should have the Data Currentness of a maximum of 30 seconds.

(R) The UME should support event notifications (i.e., SNMP Traps) for generic SNMP events (e.g., coldStart) within 2 seconds of the event detection by the UME.

4.7.7 Link Management Procedures

The network-side UME shall declare the UNI to be down either due to a physical layer failure or due to a loss of ILMI connectivity, but not due to any change in the status of the Signaling AAL. To detect loss of ILMI connectivity, the network-side UME issues periodic polls; that is, the network-side UME issues an ILMI request message every T seconds. The UNI is declared down when no ILMI response messages are received for K consecutive polls. The default values are K=4 and T=5; different values can be configured.

Link management procedures for the user-side UME are for further study.