

Network Working Group  
Request for Comments: 2320  
Category: Standards Track

M. Greene  
Xedia Corp.  
J. Luciani  
Bay Networks, Inc.  
K. White  
IBM Corp.  
T. Kuo  
Bay Networks, Inc.  
April 1998

Definitions of Managed Objects for  
Classical IP and ARP Over ATM Using SMIV2  
(IPOA-MIB)

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.

Copyright Notice

Copyright (C) The Internet Society (1998). All Rights Reserved.

Abstract

The purpose of this memo is to define the Management Information Base (MIB) for supporting Classical IP and ARP over ATM as specified in Classical IP and ARP over ATM, refer to reference [3]. Support of an ATM interface by an IP layer will require implementation of objects from several Management Information Bases (MIBs) as well as their enhancement in order to enable usage of ATM transports. It is the intent of this MIB to fully adhere to all prerequisite MIBs unless explicitly stated. Deviations will be documented in corresponding conformance statements. The specification of this MIB will utilize the Structure of Management Information (SMI) for Version 2 of the Simple Network Management Protocol Version (refer to RFC 1902, reference [1]).

## Table of Contents

1. Introduction.....	2
2. The SNMPv2 Network Management Framework.....	3
2.1 Object Definitions.....	4
3. Structure of the MIB.....	4
3.1 Basic Support MIB Definitions.....	5
3.1.1 ATM Logical IP Subnet (LIS) Table.....	5
3.1.2 ATM Logical IP Subnet Interface Mapping Table.....	7
3.1.3 ATMARP Remote Server Table.....	7
3.1.4 ATM VC Table.....	8
3.1.5 ATM Config PVC Table.....	9
3.1.6 Notifications.....	10
3.2 Client Supported MIB Definitions.....	10
3.2.1 ATMARP Client Table.....	11
3.3 Server Supported MIB Definitions.....	12
3.3.1 ATMARP Server Table.....	12
3.3.2 Notifications.....	13
4. Definitions.....	14
5. Security Considerations.....	48
6. Intellectual Property.....	49
7. Acknowledgments.....	49
8. References.....	50
9. Authors' Addresses.....	51
10. Full Copyright Statement.....	52

## 1. Introduction

This document is a product of the Internetworking Over NBMA Working Group. Its purpose is to define a MIB module for extending the traditional MIBs supported by a TCP/IP implementation to support Classical IP and ARP over ATM.

Many MIB related RFCs and Internet Drafts have been considered in the development of this document. The ones that are considered central to the extensions defined by this document are:

- o RFC 2011 - SNMPv2 Management Information Base for the Internet Protocol using SMIV2 [9]. The IP over ATM (IPOA) MIB provides extensions to the IP Group for handling IP over ATM flows. A basic understanding of the IP Group is essential for understanding this document.

- o RFC 2233 - The Interfaces Group MIB (IF-MIB) using SMIV2, reference [2]. This document is important since it provides several very useful enhancements over the interface group defined in RFC 1213 (reference [5]) that aid in handling ATM related interfaces.
- o RFC 1695 - Definitions of Managed Objects for ATM Management [4] (ATM-MIB). Support of this MIB is REQUIRED for implementing the layers between AAL5 and ATM. The contents of this MIB will not explicitly be addressed here. The ATM-MIB provides a basis for managing ATM interface layering and management of:
  - ATM Switched Virtual Connections (SVCs)
  - ATM Permanent Virtual Connections (PVCs)

The ATM Forum UNI ILMI MIB is specified by the ATM Forum in various versions of the UNI specification. The ILMI MIBs being defined are not supported via SNMP agents but via SNMP requests sent over an ATM network to an ATM entity encapsulated in an AAL5 header. Support of the ILMI MIB(s) is considered out of the scope of this document.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119, reference [10].

## 2. The SNMPv2 Network Management Framework

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

- o RFC 1902 [1] which defines the SMI, the mechanisms used for describing and naming objects for the purpose of management.
- o RFC 1903 [6] defines textual conventions for SNMPv2.
- o RFC 1904 [8] defines conformance statements for SNMPv2.
- o RFC 1905 [7] defines transport mappings for SNMPv2.
- o RFC 1906 [12] defines the protocol operations used for network access to managed objects.
- o RFC 1907 [13] defines the Management Information Base for SNMPv2.
- o RFC 1908 [14] specifies coexistence between SNMPv1 and SNMPv2.

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

This memo specifies a MIB module that is compliant to the SNMPv2 SMI. A semantically identical MIB conforming to the SNMPv1 SMI can be produced through the appropriate translation.

## 2.1. 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 refer to the object type.

## 3. Structure of the MIB

The Classical ARP and IP over ATM (IPOA) MIB structure is split into three components:

- o Basic Support MIB Definitions
- o Client Supported MIB Definitions
- o Server Supported MIB Definitions

All IP and ARP over ATM entities, both clients and ATMARP Servers, are REQUIRED to support the MIB definitions in the Basic Support MIB Definitions section. Clients need to additionally support the MIB definitions outlined in the Client specific section and ATMARP Servers MUST additionally support the ATMARP Server specific MIB definitions.

Implementation of the Definitions of Managed Objects for ATM Management [4] defines the modeling of the various layers within an ATM Interface. This modeling is assumed as a prerequisite for the IPOA-MIB. The IPOA-MIB makes no assumptions on how this layering is actually implemented within a system. Several of the MIB tables defined by the IPOA-MIB, like the base TCP/IP MIBs, require that an ifIndex exist that points to an ATM Interface. Refer to the ATM-MIB [4] for the definition of ATM Interface layering.

The use of an IP over ATM Virtual Interface layer is NOT explicitly REQUIRED by the IPOA-MIB. The use of virtual layers above an ATM-MIB defined interface layer is not absolutely necessary for modeling the

attachment of IP to an ATM network. The IPOA-MIB refers to use of a generic ifIndex object, whose value SHOULD reflect that of some specific ATM related interface as determined by an implementation. It is up to the implementers of this MIB to determine their own ATM interface layering (assuming compliance with the IF-MIB and the ATM-MIB).

The Internet Assigned Numbers Authority (IANA) ifType ipOverAtm(114) was created for use by systems that require a virtual IP over ATM interface layer. The IF-MIB's ifStackTable SHOULD be used to show the relationship between virtual IP over ATM interfaces and the actual ATM physical interface layers. The current set of ifType values can be accessed via the IANA homepage at: "<http://www.iana.org/iana/>".

### 3.1. Basic Support MIB Definitions

Basic support that MUST be implemented by both Clients and ATMARP Servers consists of:

- o ATM Logical IP Subnet (LIS) Table
- o ATM Logical IP Subnet Interface Mapping Table
- o ATMARP Remote Server Table
- o ATM VC Table
- o ATM Config PVC Table
- o Notifications

#### 3.1.1. ATM Logical IP Subnet (LIS) Table

The ATM Logical IP Subnet (LIS) Table defines the subnets that this system is a member of for purposes of reaching destinations over an ATM transport. The LIS table is indexed by the subnet address (ipoaLisSubnetAddr) and not ifIndex. The ipoaLisIfMappingTable described in the next section provides the mapping between Logical IP Subnets and the interface layer. It is possible that the same LIS can be reached via different ATM interfaces.

The ipAddrTable and the ipoaClientTable provides the mapping from a local IP address to an ATM interface. One or more ipAddrTable entries can point to the same ipoaLisEntry. An ipAddrEntry's ipAdEntAddr ANDed with its ipAdEntNetMask SHOULD equal an ipoaLisEntry's ipoaLisSubnetAddr. Given that an interface can be multi-homed, each local IP address associated with an interface requires an entry in the ipAddrTable. Each ipAddrTable entry for a local IP address associated with an ATM interface SHOULD map to an entry in the ipoaLisTable.

The bulk of the objects in an `ipoaLisEntry` exists to control ATMARP for a particular LIS. In a PVC only environment it is implementation dependent as to whether this table should be supported:

```
ipoaLisInactivityTimer
ipoaLisMinHoldingTime
ipoaLisQDepth
ipoaLisMaxCalls
ipoaLisCacheEntryAge
ipoaLisRetries
ipoaLisTimeout
```

The value of an `ipoaLisMaxCalls` object defines the maximum number of VCs that can be established simultaneously per LIS. The value of an `ipoaLisDefaultPeakCellRate` object defines the best effort default peak cell rate in both the forward and backward directions when establishing VCCs (Virtual Channel Connections). Refer to RFC 1755, ATM Signaling Support for IP over ATM (reference [11]), for a definition of the use of this object's value.

The `ipAddrTable`'s `ipAdEntReasmMaxSize` is the "The size of the largest IP datagram which this entity can re-assemble from incoming IP fragmented datagrams received on this interface" and is different from the `ipoaLisTable`'s `ipoaLisDefaultMtu` with is the default MTU used within an LIS. Note that this is the default MTU, not the actual MTU (which is represented as `ipoaVcNegotiatedMtu` in the `ipoaVcTable`).

The `ipoaLisRowStatus` object enables entries in the `ipoaLisTable` to be created or deleted via SNMP. Creation of an `ipoaLisTable` entry results in the addition of a corresponding `ipAddrTable` entry and an `ipoaLisIfMappingTable` entry. Creation of multiple `ipAddrTable` entries and `ipoaLisIfMappingTable` entries for the same LIS is not addressed by this document. When `ipoaLisRowStatus` is changed from `active(1)` to `notInService(2)` or from `active(1)` to `destroy(6)`, this has the side-effect of removing all entries from the `ipNetToMediaTable` that are associated with this LIS (in other words, it flushes the entity's ATMARP cache). It also removes the `ipoaVcTable` entries that were associated with those `ipNetToMediaTable` entries. Destroying the row removes the corresponding entries in the `ipoaArpSrvrTable`, `ipoaArpClientTable`, `ipoaLisIfMappingTable`, and the `ipoaArpRemoteSrvrTable`.

Entries in both the `ipNetToMediaTable` and the `ipoaVcTable` that are associated with an `ipoaConfigPvcEntry` are not affected by changes to `ipoaLisRowStatus`.

### 3.1.2. ATM Logical IP Subnet Interface Mapping Table

The `ipoaLisIfMappingTable` maps a LIS to all ATM interfaces from which it is configured to be supported. Each entry in the `ipoaLisIfMappingTable` SHOULD map to an `ipAddrTable` entry. It is also possible for a system, most commonly a switch, to have multiple LISs associated with the same ATM interface.

### 3.1.3. ATMARP Remote Server Table

Entries in the `ipoaArpRemoteSrvrTable` exists to locally configure the remote ATMARP Servers that exist on a per LIS and interface basis. Classical IP and ARP over ATM [3] requires that at least one ATMARP Server be configured per LIS where SVC traffic is intended. PVC usage doesn't require use of ATMARP. No `ipoaArpRemoteSrvrTable` entries SHOULD be configured for a LIS where only PVCs will be used. An entry in the `ipoaArpRemoteSrvrTable` is indexed by the subnet address of the LIS (`ipoaLisSubnetAddr`), the ATM address of the remote ATMARP Server (`ipoaArpRemoteSrvrAtmAddr`) and an interface `ifIndex` (`ipoaArpRemoteSrvrIfIndex`) value.

The object `ipoaArpRemoteSrvrIpAddr` in an `ipoaArpRemoteSrvrEntry` is set with the IP Address of the Remote ATMARP Server when a VC to the Remote ATMARP Server is established. A value of 0.0.0.0 SHOULD be used when the IP address of the Remote ATMARP Server is not known. Once `ipoaArpRemoteSrvrIpAddr` is set then the `ipoaVcTable` can be searched using `ipoaArpRemoteSrvrIfIndex` and `ipoaArpRemoteSrvrIpAddr` to find the VC in use to the Remote ATMARP Server.

`ipoaArpRemoteSrvrIfIndex` is defined to have the textual convention of `InterfaceIndexOrZero`. Adding `ipoaArpRemoteSrvrIfIndex` to the index clause allows a system to have a VC to a ATMARP Remote Server on a per LIS and interface basis. An entry in this table SHOULD exist for each interface on a per LIS basis. Each interface would then have a separate VC to the Remote ATMARP Server for ATMARP purposes.

An implementation that wants to use a single VC MAY use an `ipoaArpRemoteSrvrIfIndex` value of 0 when configuring an `ipoaArpRemoteSrvrEntry` for the associating LIS. If `ipoaArpRemoteSrvrIfIndex` is 0 then an implementation dependent method MAY be used for finding the VPI and VCI of the VC in use to the Remote ATMARP Server. For example, search the `ipoaVcTable` for a match between `ipNetToMediaNetAddress` and `ipoaArpRemoteSrvrIpAddr` from an `ipoaArpRemoteSrvrEntry`, ignoring `ipNetToMediaIfIndex`. Since a single VC is being used the first match SHOULD correspond to the correct VC.

If a PVC is intended to be used to communicate with a remote ATMARP Server then the `ipoaConfigPvcTable` MUST be used to create and activate the PVC prior to activating a `ipoaArpRemoteSrvrEntry`.

The object `ipoaArpRemoteSrvrRowStatus` allows for row creation and deletion of entries in the `ipoaArpRemoteSrvrTable`. The objects `ipoaArpRemoteSrvrAdminStatus` and `ipoaArpRemoteSrvrOperStatus` exist to control and reflect the operational use of a Remote ATMARP Server defined by an `ipoaArpRemoteSrvrEntry`. The object `ipoaArpRemoteSrvrOperStatus` SHOULD have a value of `up(1)` when an SVC has been established to the Remote ATMARP Server or if using a PVC when the InATMARP reply with the IP Address of the Remote ATMARP Server has been received. The value of `down(2)` SHOULD be used to indicate that a VC to the Remote ATMARP Server doesn't exist.

#### 3.1.4. ATM VC Table

An entry in the `ipoaVcTable` SHOULD have at least one corresponding `ipNetToMediaTable` entry. Both tables use the `ipNetToMediaTable`'s indexes `ipNetToMediaIfIndex` and `ipNetToMediaNetAddress`. The `ipoaVcTable` has the additional indexes `ipoaVcVpi` and `ipoaVcVci`. An `ipoaVcEntry` exists for every VC per ATM interface per destination IP address. Refer to the following diagram that illustrates the relationship between `ipoaVcTable` and the `ipNetToMediaTable`:

ipoaVcTable	ipNetToMediatable
ipNetToMediaIfIndex	ipNetToMediaIfIndex
ipNetToMediaNetAddress	ipNetToMediaNetAddress
ipoaVcVpi	
ipoaVcVci	
ipoaVcType	
---> use IpoaAtmAddr TC	ipNetToMediaPhysAddress
ipoaVcNegotiatedEncapsType	
ipoaVcNegotiatedMtu	ipNetToMediaType

`ipoaVcType` indicates if the entry is for an SVC or a PVC. An `ipoaVcEntry`, corresponding to an PVC, is created automatically when an `ipoaConfigPvcEntry` is created and the IP Address at the end of the PVC is discovered. The associating `ipNetToMediaTable` entry would have its `ipNetToMediaType` set to `static(4)`. `ipNetToMediaTable` entries created during ATMARP processing have a `ipNetToMediaType` of `dynamic(3)`. The process to locally configuring an `ipNetToMediaTable` entry and an `ipoaVcTable` entry for an SVC without using ATMARP is not within the scope of this document.



The objects `ipoaVcVpi` and `ipoaVcVci` are defined to have a MAX-ACCESS of not-accessible since they are only used for purposes of indexing an entry in the `ipoaVcTable`.

### 3.1.5. ATM Config PVC Table

An entry in the `ipoaVcTable` is created after the `InATMARP` reply is successfully received for an `ipoaConfigPvcEntry` during its activation. `InATMARP` should return the IP Address of the other end of the PVC in order to have the needed indexes to create an `ipNetToMediaEntry` and an `ipoaVcEntry`.

The corresponding ARP Cache entry SHOULD be deleted whenever a PVC becomes unusable.

A Network Management Station wanting to create a PVC at a particular system for use as an IP transport would:

- o use the ATM-MIB, reference [4], to create the PVC
- o use the `ipoaConfigPvcTable` in the IPOA-MIB to configure the PVC for use by IP

Refer to the following diagram that illustrates the relationship between the `ipoaVcTable` and the `ipoaConfigPvcTable`:

<code>ipoaVcTable</code>	<code>ipoaConfigPvcTable</code>
<div> <div>ipNetToMediaIfIndex</div> <div>ipNetToMediaNetAddress</div> <div>ipoaVcVpi</div> <div>ipoaVcVci</div> <div>ipoaVcType</div> <div>ipoaVcNegotiatedEncapsType</div> <div>ipoaVcNegotiatedMtu</div> </div>	<div> <div>ipNetToMediaIfIndex</div> <div>ipoaConfigPvcVpi</div> <div>ipoaConfigPvcVci</div> <div>ipoaConfigPvcDefaultMtu</div> <div>ipoaConfigPvcRowStatus</div> </div>

When the `ipoaVcEntry` is created its `ipoaVcType` will be set to `pvc(1)`, its `ipoaVcNegotiatedEncapsType` set to `llcSnap(1)`, and its `ipoaVcNegotiatedMtu` set to 9180 octets by default. Classical IP and ARP over ATM [3] allows use of other MTU values for PVCs but considers the selection of a value other than 9180 to be out of scope. `ipoaConfigPvcDefaultMtu` can be used to configure the MTU to be used for the PVC. Both ends MUST have the same value configured. The associating `ipNetToMediaTable` entry would have its `ipNetToMediaType` set to `static(4)`.

Changing `ipoaConfigPvcRowStatus` from `active(1)` to `notInService(2)` or from `active(1)` to `destroy(6)` has the side-effect of removing the corresponding `ipNetToMediaTable`, `ipoaVcTable`, and `ipoaConfigPvcTable` entries.

#### 3.1.6. Notifications

Both ATM clients and ATMARP Servers MUST support generation of an `ipoaMtuExceeded` notification.

#### 3.2. Client Supported MIB Definitions

The ATMARP Client Table is the only additional MIB table that a client MUST implement.

## 3.2.1. ATMARP Client Table

An entry in the `ipoaArpClientTable` SHOULD have a corresponding `ipAddrTable` entry where both are indexed by the same `ipAdEntAddr` value. Refer to the following diagram that illustrates the relationship between `ipoaArpClientTable` and `ipAddrTable` entries:

ipoaArpClientTable	ipAddrTable
<div>ipAdEntAddr</div> <div>ipoaArpClientAtmAddr</div> <div>ipoaArpClientSrvrInUse</div> <div>ipoaArpClientInArpInReqs</div> <div>ipoaArpClientInArpOutReqs</div> <div>ipoaArpClientInArpInReplies</div> <div>ipoaArpClientInArpOutReplies</div> <div>ipoaArpClientInArpInvalidInReqs</div> <div>ipoaArpClientInArpInvalidOutReqs</div> <div>ipoaArpClientArpInReqs</div> <div>ipoaArpClientArpOutReqs</div> <div>ipoaArpClientArpInReplies</div> <div>ipoaArpClientArpOutReplies</div> <div>ipoaArpClientArpInNaks</div> <div>ipoaArpClientArpOutNaks</div> <div>ipoaArpClientArpUnknownOps</div> <div>ipoaArpClientArpNoSrvrResps</div> <div>ipoaArpClientRowStatus</div>	<div>ipAdEntAddr</div> <div>ipAdEntNetMask</div> <div>ipAdEntIfIndex</div> <div>ipAdEntBcastAddr</div> <div>ipAdEntReasmMaxSize</div>

Both tables have the same index, `ipAdEntAddr`. The `ipAddrTable`'s `ipAdEntNetMask` when ANDed with its corresponding `ipAdEntAddr` yield the subnet of the LIS which can be used as an index into the `ipoaLisTable` (`ipoaLisSubnetAddr`). The `ipAddrTable`'s `ipAdEntIfIndex` points to an interface `ifTable` entry via an `ifIndex` value. The attachment point for IP into an ATM network is via an ATM interface's `ifIndex`. Each `ipoaArpClientEntry` MUST point to an ATM interface via its corresponding `ipAddrEntry`.

`ipoaArpClientAtmAddr` is the local ATM address associated with the corresponding ATM `ifTable` entry. `ipoaArpClientSrvrInUse` is the ATM address of the ATMARP Server being used for a particular client. If SVCs are not being used then the value of this object is a zero-length OCTET STRING.

It is sometimes possible for a system to have multiple IP addresses configured within the same IP subnet. The indexing of this table would seem to preclude that. However, it is possible to have additional entries in the `ipAddrTable` with the same `ifIndex` and with the same subnet address. The mechanism for adding these multiple entries to the `ipAddrTable` (which is read-only) is beyond the scope of this document.

The counter object `ipoaArpClientInArpInvalidInReqs` is "The number of times that this client detected an invalid `InATMARP` request." This object SHOULD be incremented when processing fails for an `InATMARP` request (e.g., for incorrect `InATMARP` request structure fields). The object `ipoaArpClientInArpInvalidOutReqs` is defined as "The number of times that this client did not receive an `InATMARP` reply." This is different from `ipoaArpClientArpNoSrvrResps` which counts the number of times no response was received from an `ATMARP` request.

`InATMARP` retransmission processing is not controlled by objects in the `ipoaListTable`. In general, the `ipoaListTable` objects relate to `ATMARP` Server processing. Configuration of `InATMARP` retransmission processing is considered to be implementation dependent and not defined by the IPOA-MIB.

Implementations SHOULD use local policy for defining both `InATMARP` timeout and retry count values. This policy would be expected to differ for sending an `InATMARP` Request over a PVC as opposed to an SVC. For transmission of an `InATMARP` Request over a SVC a timeout of 60 seconds with a retry count of 3 is suggested. `InATMARP` transmission over a PVC should differ since its retry limit may need to be infinite in order to ensure that `InATMARP` Request processing eventually occurs.

### 3.3. Server Supported MIB Definitions

`ATMARP` Servers MUST support:

- o `ATMARP` Server Table
- o Notifications

as defined in the following sections. This table exists only on a system where at least one `ATMARP` Server is present.

#### 3.3.1. `ATMARP` Server Table

This table defines the list of `ATMARP` Servers within a LIS. Each entry of the table defines each `ATMARP` Server's ATM address, the LIS it is a member of, and various `InATMARP` and `ATMARP` statistics.

An entry in this table provides information about an ATMARP Server within a LIS and is indexed by ipAdEntAddr (a local IP Address from an IP Address Table entry) and ipoaArpSrvrAddr (an ATM Address associated with the ATMARP Server).

Entries MAY be created by a management application using the ipoaArpSrvrRowStatus object. Entries in this table MAY also be created by the system and not by a management application, for example via ILMI.

Entries in this table MAY be deleted by setting the ipoaArpSrvrRowStatus object to destroy(6). This includes entries that were added by the system and not by a management application.

On a host that supports multiple ATMARP Servers where the local IP address being associated with each ATMARP Server is the same (for example a non-multihomed host), the ATM Address (ipoaArpSrvrAddr) uniquely identifies a particular ATMARP Server. On a host supporting multiple ATMARP Servers having a single ATM Interface with a single ATM Address, the ipAdEntAddr MUST be used to uniquely identify an entry in the ipoaArpSrvrTable.

The indexing of the ipoaArpSrvrTable does not allow entries with the same or no local IP Address (ipAdEntAddr) and the same ATM Address (ipoaArpSrvrAddr) to exist. The values of the index elements when combined to index a row must be unique.

### 3.3.2. Notifications

An ATMARP Server MUST support the following notifications:

- o ipoaDuplicateIpAddress
- o ipoaLisCreate
- o ipoaLisDelete

Generation of ipoaLisCreate and ipoaLisDelete notifications is controlled by the ipoaLisTrapEnable object. These notifications indicate when an ipoaLisEntry is either created or deleted. The purpose of these notifications is to enable Network Management Applications to dynamically discover the existence of ATMARP Server LIS participation in order to eventually determine LIS composition via subsequent SNMP queries. It is permissible for an ATM client-only system to support the ipoaLisTrapEnable object and generate ipoaLisCreate and ipoaLisDelete notifications.

## 4. Definitions

```
IPOA-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
    MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
    transmission, Integer32, IpAddress, Counter32,
    Gauge32
        FROM SNMPv2-SMI
    TEXTUAL-CONVENTION, RowStatus
        FROM SNMPv2-TC
    MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
        FROM SNMPv2-CONF
    ipNetToMediaNetAddress, ipNetToMediaIfIndex,
    ipNetToMediaPhysAddress, ipAdEntAddr
        FROM IP-MIB
```

```
-- The following textual conventions are defined locally within
-- this MIB module. They have been prefixed with 'Ipoa' to
-- distinguish them from their counterparts in the ATM-TC-MIB.
-- This was done so that the IPOA-MIB could be advanced as
-- a standards-based MIB without waiting for the ATM-TC-MIB.
```

```
-- AtmConnKind, AtmAddr
--     FROM ATM-TC-MIB
```

```
    InterfaceIndex, InterfaceIndexOrZero
        FROM IF-MIB
;
```

```
ipoaMIB MODULE-IDENTITY
```

```
    LAST-UPDATED "9802090000Z" -- February 9, 1998
    ORGANIZATION "IETF Internetworking Over NBMA Working
        Group (ion)"
```

```
    CONTACT-INFO
```

```
        "Maria Greene (greene@xedia.com)
        Xedia Corp.
```

```
        Jim Luciani (jluciani@BayNetworks.com)
        Bay Networks
```

```
        Kenneth White (kennethw@vnet.ibm.com)
        IBM Corp.
```

```
        Ted Kuo (tkuo@eos.ncsu.edu)
        Bay Networks"
```

```
    DESCRIPTION
```

```
        "This module defines a portion of the management
```

information base (MIB) for managing Classical IP and  
 ARP over ATM entities."  
 ::= { transmission 46 }

## -- Textual Conventions

IpoaEncapsType ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The encapsulation type used on a VC."

SYNTAX INTEGER {  
     llcSnap(1),  
     vcMuxed(2),  
     other(3)  
 }

IpoaVpiInteger ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"An integer large enough to contain the value of a VPI."

SYNTAX Integer32 (0..255)

IpoaVciInteger ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"An integer large enough to contain the value of a VCI."

SYNTAX Integer32 (0..65535)

IpoaAtmAddr ::= TEXTUAL-CONVENTION

DISPLAY-HINT "1x"

STATUS current

DESCRIPTION

"The ATM address used by the network entity.  
 The semantics are implied by the length.  
 The address types are:

- no address (0 octets)
- E.164 (8 octets)
- NSAP (20 octets)

In addition, when subaddresses are used IpoaAtmAddr  
 may represent the concatenation of address and  
 subaddress. The associated address types are:

- E.164, E.164 (16 octets)
- E.164, NSAP (28 octets)
- NSAP, NSAP (40 octets)

Address lengths other than defined in this definition imply address types defined elsewhere.

Note: The E.164 address is encoded in BCD format."

SYNTAX OCTET STRING (SIZE(0..40))

IpoaAtmConnKind ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The use of call control. The use is as follows:

pvc(1)

Virtual link of a PVC. Should not be used in a PVC/SVC (i.e., SPVC) crossconnect.

svcIncoming(2)

Virtual link established after a received signaling request to setup an SVC.

svcOutgoing(3)

Virtual link established after a transmitted or forwarded signaling request to setup an SVC.

spvcInitiator(4)

Virtual link at the PVC side of an SVC/PVC crossconnect, where the switch is the initiator of the SPVC setup.

spvcTarget(5)

Virtual link at the PVC side of an SVC/PVC crossconnect, where the switch is the target of the SPVC setup.

An spvcInitiator is always cross-connected to an svcOutgoing, and an spvcTarget is always cross-connected to an svcIncoming."

```
SYNTAX  INTEGER {
    pvc(1),
    svcIncoming(2),
    svcOutgoing(3),
    spvcInitiator(4),
    spvcTarget(5)
}
```

-- Top-level structure of the MIB

```
ipoaObjects      OBJECT IDENTIFIER ::= { ipoaMIB 1 }
ipoaNotifications OBJECT IDENTIFIER ::= { ipoaMIB 2 }
ipoaConformance  OBJECT IDENTIFIER ::= { ipoaMIB 3 }
```



-- MIB Objects

ipoaLisTrapEnable OBJECT-TYPE  
SYNTAX INTEGER { enabled(1), disabled(2) }  
MAX-ACCESS read-write  
STATUS current  
DESCRIPTION  
"Indicates whether ipoaLisCreate and ipoaLisDelete  
traps should be generated by this system.  
  
By default, this object should have the value  
enabled(1) for systems where ATMARP Servers are  
present and disabled(2) on systems where only  
clients reside."  
::= { ipoaObjects 1 }

-- The ATM Logical IP Subnet (LIS) Table

ipoaLisTable OBJECT-TYPE  
SYNTAX SEQUENCE OF IpoaLisEntry  
MAX-ACCESS not-accessible  
STATUS current  
DESCRIPTION  
"There is one entry in this table for every Logical IP  
Subnet (LIS) of which this system is a member.  
  
The bulk of the objects in an ipoaLisEntry exists  
to control ATMARP for a particular LIS. In a PVC only  
environment it is implementation dependent as to  
whether this table should be supported."  
::= { ipoaObjects 2 }

ipoaLisEntry OBJECT-TYPE  
SYNTAX IpoaLisEntry  
MAX-ACCESS not-accessible  
STATUS current  
DESCRIPTION  
"Information about a single LIS of which this system  
is a member.  
  
Membership in a LIS is independent of the actual ATM  
interfaces being used. The ipoaLisTable defines  
all LISs that a system is a member of. The ipAddrTable  
and the ipoaClientTable provides the mapping from local  
IP address to ATM interface. The ipoaLisIfMappingTable  
provides the mappings between Logical IP Subnets and  
interfaces."

The ipoaLisTable is indexed by ipoaLisSubnetAddr (IP subnet address). An entry in the ipoaLisTable should exist for each ipAddrEntry that is associated with an ATM related interface used for Classical IP and ARP over ATM traffic.

Its ipAdEntAddr and ipAdEntNetMask when ANDed together should equal the ipoaLisSubnetAddr of the corresponding ipoaLisEntry."

```
INDEX      { ipoaLisSubnetAddr }
 ::= { ipoaLisTable 1 }
```

```
IpoaLisEntry ::= SEQUENCE {
    ipoaLisSubnetAddr      IpAddress,
    ipoaLisDefaultMtu      Integer32,
    ipoaLisDefaultEncapsType IpoaEncapsType,
    ipoaLisInactivityTimer Integer32,
    ipoaLisMinHoldingTime  Integer32,
    ipoaLisQDepth          Integer32,
    ipoaLisMaxCalls        Integer32,
    ipoaLisCacheEntryAge   Integer32,
    ipoaLisRetries         Integer32,
    ipoaLisTimeout         Integer32,
    ipoaLisDefaultPeakCellRate Integer32,
    ipoaLisActiveVcs       Gauge32,
    ipoaLisRowStatus       RowStatus
}
```

ipoaLisSubnetAddr OBJECT-TYPE

```
SYNTAX      IpAddress
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The IP subnet address associated with this LIS."
 ::= { ipoaLisEntry 1 }
```

ipoaLisDefaultMtu OBJECT-TYPE

```
SYNTAX      Integer32 (0..65535)
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "The default MTU used within this LIS. Note that the
    actual MTU used for a VC between two members of the
    LIS may be negotiated during connection setup and may
    be different than this value. The ipoaVcNegotiatedMtu
    object indicates the actual MTU in use for a
    particular VC."
DEFVAL { 9180 }
```

```
::= { ipoaLisEntry 2 }
```

#### ipoaLisDefaultEncapsType OBJECT-TYPE

SYNTAX IpoaEncapsType

MAX-ACCESS read-create

STATUS current

##### DESCRIPTION

"The default encapsulation to use on VCs created for this LIS. Note that the actual encapsulation type may be negotiated during connection setup and may be different than this value. The ipoaVcNegotiatedEncapsType object indicates the actual encapsulation in use for a particular VC."

DEFVAL { llcSnap }

```
::= { ipoaLisEntry 3 }
```

#### ipoaLisInactivityTimer OBJECT-TYPE

SYNTAX Integer32

UNITS "seconds"

MAX-ACCESS read-create

STATUS current

##### DESCRIPTION

"The time, in seconds, before a call established for an ipNetToMediaEntry on a client will timeout due to no traffic being passed on the VC. A value of 0 implies no time out."

##### REFERENCE

"RFC 1755, Sec. 3.4 VC Teardown"

DEFVAL { 1200 }

```
::= { ipoaLisEntry 4 }
```

#### ipoaLisMinHoldingTime OBJECT-TYPE

SYNTAX Integer32 (0..65535)

UNITS "seconds"

MAX-ACCESS read-create

STATUS current

##### DESCRIPTION

"The minimum amount of time, in seconds, that a call will remain open. If 0 then ipoaInactivityTimer will completely determine when a call is terminated."

##### REFERENCE

"RFC 1755, Sec. 3.4 VC Teardown"

DEFVAL { 60 }

```
::= { ipoaLisEntry 5 }
```

#### ipoaLisQDepth OBJECT-TYPE

SYNTAX Integer32 (1..65535)

UNITS "packets"

```

MAX-ACCESS    read-create
STATUS        current
DESCRIPTION
    "The maximum number of outstanding requests that are
    allowed while waiting for ATMARP replies and
    InATMARP replies for this LIS."
DEFVAL        { 1 }
::= { ipoaLisEntry 6 }

```

```

ipoaLisMaxCalls OBJECT-TYPE
    SYNTAX      Integer32 (1..65535)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The maximum number of SVCs that can be established
        simultaneously for this LIS."
    DEFVAL      { 500 }
    ::= { ipoaLisEntry 7 }

```

```

ipoaLisCacheEntryAge OBJECT-TYPE
    SYNTAX      Integer32 (60..1200)
    UNITS       "seconds"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The time, in seconds, before an ipNetToMediaEntry will
        age out of the table. Note that the default value will
        be different for a client and a server. An ATMARP
        Server should use a default of 1200 and a client should
        use 900."
    DEFVAL      { 900 }
    ::= { ipoaLisEntry 8 }

```

```

ipoaLisRetries OBJECT-TYPE
    SYNTAX      Integer32 (0..10)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The number of times the ATMARP request will be retried
        when no response is received in the timeout interval
        indicated by ipoaLisTimeout."
    DEFVAL      { 2 }
    ::= { ipoaLisEntry 9 }

```

```

ipoaLisTimeout OBJECT-TYPE
    SYNTAX      Integer32 (1..60)
    UNITS       "seconds"
    MAX-ACCESS  read-create

```

```

STATUS      current
DESCRIPTION
    "The time to wait, in seconds, before retransmission
    of an ARP request."
DEFVAL      { 10 }
::= { ipoaLisEntry 10 }

```

#### ipoaLisDefaultPeakCellRate OBJECT-TYPE

```

SYNTAX      Integer32
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "This object is the signalling parameter that
    should be used when setting up all best effort
    VCCs (Virtual Channel Connections).
    This parameter applies to the forward and
    backward direction on a per best effort VCC basis.
    A value of zero implies that no configured default
    exists and that local policy should be used to
    determine the actual default to used during
    call setup.  ATM Signaling Support for IP over ATM
    (RFC 1755) recommends 1/10th of the ATM interface's
    speed."
::= { ipoaLisEntry 11 }

```

#### ipoaLisActiveVcs OBJECT-TYPE

```

SYNTAX      Gauge32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "Number of active SVCs for this LIS."
::= { ipoaLisEntry 12 }

```

#### ipoaLisRowStatus OBJECT-TYPE

```

SYNTAX      RowStatus
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "This object allows entries to be created and deleted
    in the ipoaLisTable.

    When the ipoaLisRowStatus deleted (by setting this
    object to destroy(6)), this has the side-effect of
    removing all entries from the ipNetToMediaTable that
    are associated with this LIS (in other words, it
    flushes the entity's ATMARP cache).  It also removes
    the ipoaVcTable entries that were associated with those
    ipNetToMediaTable entries.  Destroying the row also

```

removes the corresponding entries in the  
 ipoaArpSrvrTable, ipoaArpClientTable,  
 ipoaLisIfMappingTable, and ipoaArpRemoteSrvrTable.

Entries in both the ipNetToMediaTable and the  
 ipoaVcTable that are associated with a  
 ipoaConfigPvcEntry are not affected by changes to  
 ipoaLisRowStatus."

#### REFERENCE

"RFC 1903, 'Textual Conventions for Version 2 of the  
 Simple Network Management Protocol (SNMPv2).'

::= { ipoaLisEntry 13 }

-- The ATM Logical IP Subnet Interface Mapping Table

ipoaLisIfMappingTable OBJECT-TYPE

SYNTAX SEQUENCE OF IpoaLisIfMappingEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"There is one entry in this table for every combination  
 of ipoaLisEntry and IP over ATM interface."

::= { ipoaObjects 3 }

ipoaLisIfMappingEntry OBJECT-TYPE

SYNTAX IpoaLisIfMappingEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Defines an entry in the ipoaLisIfMappingTable."

INDEX { ipoaLisSubnetAddr, ipoaLisIfMappingIfIndex }

::= { ipoaLisIfMappingTable 1 }

IpoaLisIfMappingEntry ::= SEQUENCE {

ipoaLisIfMappingIfIndex InterfaceIndex,

ipoaLisIfMappingRowStatus RowStatus

}

ipoaLisIfMappingIfIndex OBJECT-TYPE

SYNTAX InterfaceIndex

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The ipAdEntIfIndex object from an ipAddrEntry  
 is used as an index to this table when its  
 ipAdEntAddr is in the subnet implied by  
 ipoaLisSubnetAddr."

```
::= { ipoaLisIfMappingEntry 1 }
```

```
ipoaLisIfMappingRowStatus OBJECT-TYPE
```

```
SYNTAX      RowStatus
MAX-ACCESS  read-create
STATUS      current
```

```
DESCRIPTION
```

```
"This object allows entries to be created and deleted
in the ipoaLisIfMappingTable."
```

```
REFERENCE
```

```
"RFC 1903, 'Textual Conventions for Version 2 of the
Simple Network Management Protocol (SNMPv2).'"
```

```
::= { ipoaLisIfMappingEntry 2 }
```

```
-- The ATMARP Client Table
```

```
ipoaArpClientTable OBJECT-TYPE
```

```
SYNTAX      SEQUENCE OF IpoaArpClientEntry
MAX-ACCESS  not-accessible
STATUS      current
```

```
DESCRIPTION
```

```
"The ATMARP clients running on this system."
```

```
::= { ipoaObjects 4 }
```

```
ipoaArpClientEntry OBJECT-TYPE
```

```
SYNTAX      IpoaArpClientEntry
MAX-ACCESS  not-accessible
STATUS      current
```

```
DESCRIPTION
```

```
"Information about a single ATMARP Client.  Clients
can be started and stopped by adding and removing
entries from this table.  An entry in the
ipoaArpClientTable has a corresponding entry in the
ipAddrTable.  Both are indexed by ipAdEntAddr.
The ifIndex and subnet mask of a client entry are the
ipAddrEntry's ipAdEntIfIndex and ipAdEntNetMask,
respectively.
```

Note that adding and removing entries from this table may have the same effect on the corresponding ipAddrTable entry. Row creation of an entry in this table requires that either the corresponding ipAddrTable entry exists or that ipAdEntIfIndex and ipAdEntNetMask be specified in the creation of an ipoaArpClientEntry at a minimum in order to create the corresponding ipAddrEntry. Specification of ipAdEntBcastAddr and ipAdEntReasmMaxSize to complete an ipAddrEntry is implementation dependent.

Whether a corresponding ipAddrEntry is deleted during the deletion of an ipoaArpClientEntry is considered implementation dependent."

```
INDEX      { ipAdEntAddr }
 ::= { ipoaArpClientTable 1 }
```

```
IpoaArpClientEntry ::= SEQUENCE {
    ipoaArpClientAtmAddr      IpoaAtmAddr,
    ipoaArpClientSrvrInUse    IpoaAtmAddr,
    ipoaArpClientInArpInReqs  Counter32,
    ipoaArpClientInArpOutReqs Counter32,
    ipoaArpClientInArpInReplies Counter32,
    ipoaArpClientInArpOutReplies Counter32,
    ipoaArpClientInArpInvalidInReqs Counter32,
    ipoaArpClientInArpInvalidOutReqs Counter32,
    ipoaArpClientArpInReqs    Counter32,
    ipoaArpClientArpOutReqs   Counter32,
    ipoaArpClientArpInReplies Counter32,
    ipoaArpClientArpOutReplies Counter32,
    ipoaArpClientArpInNaks    Counter32,
    ipoaArpClientArpOutNaks   Counter32,
    ipoaArpClientArpUnknownOps Counter32,
    ipoaArpClientArpNoSrvrResps Counter32,
    ipoaArpClientRowStatus    RowStatus
}
```

```
ipoaArpClientAtmAddr OBJECT-TYPE
    SYNTAX      IpoaAtmAddr
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "The ATM address of the client."
    ::= { ipoaArpClientEntry 1 }
```

```
ipoaArpClientSrvrInUse OBJECT-TYPE
    SYNTAX      IpoaAtmAddr
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The ATM address of the ATMARP Server,
        ipoaArpRemoteSrvrAtmAddr, in use by this client. A
        zero length octet string implies that communication
        with a Remote ATMARP Server is not in effect."
    DEFVAL { ''H }
    ::= { ipoaArpClientEntry 2 }
```

```
ipoaArpClientInArpInReqs OBJECT-TYPE
    SYNTAX      Counter32
```



MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
    "The number of InATMARP requests received by this  
    client."  
::= { ipoaArpClientEntry 3 }

ipoaArpClientInArpOutReqs OBJECT-TYPE  
SYNTAX Counter32  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
    "The number of InATMARP requests sent by this client."  
::= { ipoaArpClientEntry 4 }

ipoaArpClientInArpInReplies OBJECT-TYPE  
SYNTAX Counter32  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
    "The number of InATMARP replies received by this  
    client."  
::= { ipoaArpClientEntry 5 }

ipoaArpClientInArpOutReplies OBJECT-TYPE  
SYNTAX Counter32  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
    "Total number of InATMARP replies sent by this client."  
::= { ipoaArpClientEntry 6 }

ipoaArpClientInArpInvalidInReqs OBJECT-TYPE  
SYNTAX Counter32  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
    "The number of times that this client detected an  
    invalid InATMARP request."  
::= { ipoaArpClientEntry 7 }

ipoaArpClientInArpInvalidOutReqs OBJECT-TYPE  
SYNTAX Counter32  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
    "The number of times that this client did not  
    receive an InATMARP reply."

```
::= { ipoaArpClientEntry 8 }

ipoaArpClientArpInReqs OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Total number of ATMARP requests received by this
        client."
    ::= { ipoaArpClientEntry 9 }

ipoaArpClientArpOutReqs OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Total number of ATMARP requests sent by this client."
    ::= { ipoaArpClientEntry 10 }

ipoaArpClientArpInReplies OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Total number of ATMARP replies received by this
        client."
    ::= { ipoaArpClientEntry 11 }

ipoaArpClientArpOutReplies OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Total number of ATMARP replies sent by this client."
    ::= { ipoaArpClientEntry 12 }

ipoaArpClientArpInNaks OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Total number of negative ATMARP replies
        received by this client."
    ::= { ipoaArpClientEntry 13 }

ipoaArpClientArpOutNaks OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
```

STATUS current

DESCRIPTION

"Total number of negative ATMARP replies sent by this client.

Classic IP and ARP over ATM does not require an ATMARP client to transmit an ATMARP\_NAK upon receipt of an ATMARP request from another ATMARP client. However, implementation experience has shown that this error condition is somewhat easy to create inadvertently by configuring one ATMARP client with an ipoaArpRemoteSrvrTable entry containing an ipoaArpRemoteSrvrAtmAddr value which is the ATM address of another ATMARP client-only system.

If an ATMARP client supports the transmission of ATMARP\_NAKs, then it should increment ipoaArpClientArpOutNaks each time it transmits an ATMARP\_NAK. Otherwise, support of this object is considered optional."

::= { ipoaArpClientEntry 14 }

ipoaArpClientArpUnknownOps OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of times that this client received an ATMARP message with an operation code for which it is not coded to support."

::= { ipoaArpClientEntry 15 }

ipoaArpClientArpNoSrvrResps OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of times this client failed to receive a response from a ATMARP Server within the ipoaLisTimeout value for ipoaLisRetries times. This may imply that the client will re-elect a new primary ATMARP Server for this LIS from the ipoaArpRemoteSrvrTable."

::= { ipoaArpClientEntry 16 }

ipoaArpClientRowStatus OBJECT-TYPE

SYNTAX RowStatus

```

MAX-ACCESS    read-create
STATUS        current
DESCRIPTION
    "This object allows entries to be created and
    deleted from the ipoaArpClientTable."
REFERENCE
    "RFC 1903, 'Textual Conventions for Version 2 of the
    Simple Network Management Protocol (SNMPv2).'"
::= { ipoaArpClientEntry 17 }

```

-- The ATMARP Server Table

```

ipoaArpSrvrTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF IpoaArpSrvrEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The ATMARP Servers running on this system."
    ::= { ipoaObjects 5 }

```

```

ipoaArpSrvrEntry OBJECT-TYPE
    SYNTAX      IpoaArpSrvrEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Information about an ATMARP Server within a LIS.  An
        entry in this table has two indexes: first ipAdEntAddr,
        which is the IP address that this system uses as a
        member of the LIS, and then ipoaArpSrvrAddr, which is
        the ATM address of the ATMARP Server.

```

Entries may be created by a management application using the ipoaArpSrvrRowStatus object. Entries in this table may also be created by the system and not by a management application, for example via ILMI.

Entries in this table may be deleted by setting the ipoaArpSrvrRowStatus object to 'destroy(6)'. This includes entries that were added by the system and not by a management application."

```

INDEX { ipAdEntAddr, ipoaArpSrvrAddr }
::= { ipoaArpSrvrTable 1 }

```

```

IpoaArpSrvrEntry ::= SEQUENCE {
    ipoaArpSrvrAddr      IpoaAtmAddr,
    ipoaArpSrvrLis       IpAddress,
    ipoaArpSrvrInArpInReqs Counter32,
    ipoaArpSrvrInArpOutReqs Counter32,

```

```

        ipoaArpSrvrInArpInReplies      Counter32,
        ipoaArpSrvrInArpOutReplies     Counter32,
        ipoaArpSrvrInArpInvalidInReqs  Counter32,
        ipoaArpSrvrInArpInvalidOutReqs Counter32,
        ipoaArpSrvrArpInReqs           Counter32,
        ipoaArpSrvrArpOutReplies       Counter32,
        ipoaArpSrvrArpOutNaks          Counter32,
        ipoaArpSrvrArpDupIpAddrs       Counter32,
        ipoaArpSrvrArpUnknownOps       Counter32,
        ipoaArpSrvrRowStatus            RowStatus
    }

ipoaArpSrvrAddr OBJECT-TYPE
    SYNTAX      IpoaAtmAddr
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The ATM address of the ATMARP Server."
    ::= { ipoaArpSrvrEntry 1 }

ipoaArpSrvrLis OBJECT-TYPE
    SYNTAX      IpAddress
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The subnet address that identifies the LIS with
         which this server is associated."
    ::= { ipoaArpSrvrEntry 2 }

ipoaArpSrvrInArpInReqs OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of InATMARP requests received by this
         ATMARP Server."
    ::= { ipoaArpSrvrEntry 3 }

ipoaArpSrvrInArpOutReqs OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of InATMARP requests sent by this ATMARP
         Server."
    ::= { ipoaArpSrvrEntry 4 }

ipoaArpSrvrInArpInReplies OBJECT-TYPE

```

```
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of InATMARP replies received by this
    ATMARP Server."
 ::= { ipoaArpSrvrEntry 5 }
```

ipoaArpSrvrInArpOutReplies OBJECT-TYPE

```
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of InATMARP replies sent by this ATMARP
    Server."
 ::= { ipoaArpSrvrEntry 6 }
```

ipoaArpSrvrInArpInvalidInReqs OBJECT-TYPE

```
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of invalid InATMARP requests received by
    this ATMARP Server."
 ::= { ipoaArpSrvrEntry 7 }
```

ipoaArpSrvrInArpInvalidOutReqs OBJECT-TYPE

```
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of times that this server did not receive
    an InATMARP reply."
 ::= { ipoaArpSrvrEntry 8 }
```

ipoaArpSrvrArpInReqs OBJECT-TYPE

```
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "Total number of ATMARP requests received by this
    ATMARP Server."
 ::= { ipoaArpSrvrEntry 9 }
```

ipoaArpSrvrArpOutReplies OBJECT-TYPE

```
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
```

## DESCRIPTION

"Total number of ATMARP replies sent by this ATMARP Server."

::= { ipoaArpSrvrEntry 10 }

## ipoaArpSrvrArpOutNaks OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"Total number of negative ATMARP replies sent by this ATMARP Server."

::= { ipoaArpSrvrEntry 11 }

## ipoaArpSrvrArpDupIpAddrs OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The number of times that a duplicate IP address was detected by this ATMARP Server."

::= { ipoaArpSrvrEntry 12 }

## ipoaArpSrvrArpUnknownOps OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The number of times that this ATMARP Server received an ATMARP message with an operation code for which it is not coded to support."

::= { ipoaArpSrvrEntry 13 }

## ipoaArpSrvrRowStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

## DESCRIPTION

"This object allows entries to be created and deleted from the ipoaArpSrvrTable."

## REFERENCE

"RFC 1903, 'Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2).'"

::= { ipoaArpSrvrEntry 14 }

## -- The Remote ATMARP Server Table

## ipoaArpRemoteSrvrTable OBJECT-TYPE

SYNTAX SEQUENCE OF IpoaArpRemoteSrvrEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A table of non-local ATMARP Servers associated with a LIS. An entry in this table has three indexes: first the ipoaLisSubnetAddr of the LIS for which the corresponding ATMARP Server provides ATMARP services, then the ipoaArpRemoteSrvrAtmAddr, which is the ATM address of the remote ATMARP Server, and finally the ifIndex of the interface on which the VC to the ATMARP Remote Server will be opened. An ifIndex value of 0 should be used when a single VC is to be shared for ATMARP purposes by multiple interfaces."

::= { ipoaObjects 6 }

ipoaArpRemoteSrvrEntry OBJECT-TYPE

SYNTAX IpoaArpRemoteSrvrEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Information about one non-local ATMARP Server."

INDEX { ipoaLisSubnetAddr, ipoaArpRemoteSrvrAtmAddr,  
ipoaArpRemoteSrvrIfIndex }

::= { ipoaArpRemoteSrvrTable 1 }

IpoaArpRemoteSrvrEntry ::= SEQUENCE {

ipoaArpRemoteSrvrAtmAddr IpoaAtmAddr,

ipoaArpRemoteSrvrRowStatus RowStatus,

ipoaArpRemoteSrvrIfIndex InterfaceIndexOrZero,

ipoaArpRemoteSrvrIpAddr IpAddress,

ipoaArpRemoteSrvrAdminStatus INTEGER,

ipoaArpRemoteSrvrOperStatus INTEGER

}

ipoaArpRemoteSrvrAtmAddr OBJECT-TYPE

SYNTAX IpoaAtmAddr

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The ATM address of the remote ATMARP Server."

::= { ipoaArpRemoteSrvrEntry 1 }

ipoaArpRemoteSrvrRowStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION



"This object allows entries to be created and deleted from the ipoaArpRemoteSrvrTable.

Deleting an ipoaArpRemoteSrvrEntry (by setting this object to destroy(6)) may affect ipoaArpClientTable entries. The object ipoaArpClientSrvrInUse in an ipoaArpClientSrvrEntry may contain the ATM address of an ATMARP Remote Server whose entry in the ipoaArpRemoteSrvrTable is being removed. In this case, any corresponding ipoaArpClientSrvrInUse objects should be at a minimum invalidated by setting their values to that of a zero length OCTET STRING.

The value of ipoaArpRemoteSrvrOperStatus should be consistent with that of ipoaArpRemoteSrvrRowStatus. For example, successfully setting the value of this object to notInService(2) after its being in the up(1) state should result in ipoaArpRemoteSrvrOperStatus being set to down(2) if currently up(1)."

#### REFERENCE

"RFC 1903, 'Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2).'

::= { ipoaArpRemoteSrvrEntry 2 }

#### ipoaArpRemoteSrvrIfIndex OBJECT-TYPE

SYNTAX InterfaceIndexOrZero

MAX-ACCESS not-accessible

STATUS current

#### DESCRIPTION

"The ifIndex of the interface that the VC to the Remote ATMARP Server is associated with."

::= { ipoaArpRemoteSrvrEntry 3 }

#### ipoaArpRemoteSrvrIpAddress OBJECT-TYPE

SYNTAX IpAddress

MAX-ACCESS read-only

STATUS current

#### DESCRIPTION

"The IP Address of the Remote ATMARP Server. A value of 0.0.0.0 implies that this address isn't known."

DEFVAL { '00000000'H }

::= { ipoaArpRemoteSrvrEntry 4 }

#### ipoaArpRemoteSrvrAdminStatus OBJECT-TYPE

SYNTAX INTEGER {

```

        up(1), -- use this ATMARP Server
        down(2) -- stop using this ATMARP Server
    }
MAX-ACCESS    read-create
STATUS        current
DESCRIPTION
    "The desired state for use of the ATMARP Server
    represented by an entry in this table.
    ipoaArpRemoteSrvrAdminStatus values:

    up(1)      - Attempt to activate use of the
                  ATMARP Server represented by this
                  entry in the ipoaArpRemoteSrvrTable.
    down(2)    - Deactivate use of this ATMARP
                  Server.

    When a managed system creates an entry in this
    table ipoaArpRemoteSrvrAdminStatus and
    ipoaArpRemoteSrvrOperStatus are initialized as
    down(2) by default."
DEFVAL { down }
::= { ipoaArpRemoteSrvrEntry 5 }

ipoaArpRemoteSrvrOperStatus OBJECT-TYPE
    SYNTAX     INTEGER {
        up(1), -- eligible for use
        down(2) -- not eligible for use
    }
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "The current operational state for use of a Remote
        ATMARP Server.  An up(1) entry has a VC
        established to the respective Remote ATMARP
        Server:

        up(1)    - A VC exists to Remote ATMARP Server
                    whose IP Address is stored in
                    ipoaArpRemoteSrvrIpAddr.  This VC can
                    be determined by searching the
                    ipoaVcTable using
                    ipoaArpRemoteSrvrIfIndex (if not 0,
                    otherwise ignore ipNetToMediaIfIndex
                    index) and ipoaArpRemoteSrvrIpAddr.
                    An ipoaArpClientEntry should exist
                    with its ipoaArpClientSrvrInUse
                    object having the same value as
                    ipoaArpRemoteSrvrAtmAddr.
```

down(2) - Entry exists without an active VC to the Remote ATMARP Server.

Transition from up(1) to down(2) status may affect ipoaArpClientTable entries. The object ipoaArpClientSrvrInUse in an ipoaArpClientSrvrEntry may contain the ATM address of an ATMARP Remote Server whose entry in the ipoaArpRemoteSrvrTable is being deactivated. In this case, any corresponding ipoaArpClientSrvrInUse objects should be at a minimum invalidated by setting their values to that of a zero length OCTET STRING.

If ipoaArpRemoteSrvrAdminStatus is down(2) then ipoaArpRemoteSrvrOperStatus should be down(2). If ipoaArpRemoteSrvrAdminStatus is changed to up(1) then ipoaArpRemoteSrvrOperStatus should change to up(1) if the Remote ATMARP Server entry can be activated."

```
DEFVAL { down }
 ::= { ipoaArpRemoteSrvrEntry 6 }
```

-- The ATM VC Table

ipoaVcTable OBJECT-TYPE

SYNTAX SEQUENCE OF IpoaVcEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A system that supports IP over ATM is an IP system and therefore MUST support all of the appropriate tables in the SNMPv2-MIB (RFC 1907), the IF-MIB (RFC 2233), the IP-MIB (RFC 2011), the TCP-MIB (RFC 2012), and the UDP-MIB (RFC 2013). This includes the ipNetToMediaTable (the ARP cache) that is defined within the IP-MIB (RFC 2011). The ipoaVcTable keeps a set of VCs for each entry in the ARP cache that was put there by an IP over ATM system acting as either a host or server. The ipoaVcTable doesn't augment the ipNetToMediaTable (ARP Cache) since the correspondence between tables is not necessarily one-to-one.

An ipNetToMediaPhysAddress object should contain the content as defined by the IpoaAtmAddr textual convention when used to hold an IPOA-MIB ATM Address."

```
::= { ipoaObjects 7 }
```

## ipoaVcEntry OBJECT-TYPE

SYNTAX IpoaVcEntry  
 MAX-ACCESS not-accessible  
 STATUS current

## DESCRIPTION

"A VC (permanent or switched) that this host or server has opened with another member of a LIS. Additional information can be determined about the VC from the ATM-MIB.

Entries in this table cannot be created by management applications.

In an SVC environment, an entry is automatically added by the system as the result of ATMARP processing.

In a PVC environment, an entry is automatically added to this table when an entry is created in the ipoaConfigPvcTable and the IP Address at the remote end of the PVC is discovered using InATMARP. An entry also is added to the ipNetToMediaTable."

INDEX { ipNetToMediaIfIndex,  
           ipNetToMediaNetAddress,  
           ipoaVcVpi,  
           ipoaVcVci  
       }  
 ::= { ipoaVcTable 1 }

IpoaVcEntry ::= SEQUENCE {

ipoaVcVpi	IpoaVpiInteger,
ipoaVcVci	IpoaVciInteger,
ipoaVcType	IpoaAtmConnKind,
ipoaVcNegotiatedEncapsType	IpoaEncapsType,
ipoaVcNegotiatedMtu	Integer32 }

## ipoaVcVpi OBJECT-TYPE

SYNTAX IpoaVpiInteger  
 MAX-ACCESS not-accessible  
 STATUS current

## DESCRIPTION

"The VPI value for the Virtual Circuit."

::= { ipoaVcEntry 1 }

## ipoaVcVci OBJECT-TYPE

SYNTAX IpoaVciInteger  
 MAX-ACCESS not-accessible  
 STATUS current

## DESCRIPTION

"The VCI value for the Virtual Circuit."

::= { ipoaVcEntry 2 }

## ipoaVcType OBJECT-TYPE

SYNTAX IpoaAtmConnKind

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The type of the Virtual Circuit."

::= { ipoaVcEntry 3 }

## ipoaVcNegotiatedEncapType OBJECT-TYPE

SYNTAX IpoaEncapType

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The encapsulation type used when communicating over this circuit."

::= { ipoaVcEntry 4 }

## ipoaVcNegotiatedMtu OBJECT-TYPE

SYNTAX Integer32 (0..65535)

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The MTU used when communicating over this circuit."

::= { ipoaVcEntry 5 }

## -- The ATM Config PVC Table

## ipoaConfigPvcTable OBJECT-TYPE

SYNTAX SEQUENCE OF IpoaConfigPvcEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"This table MUST be supported when PVCs are intended to be supported in order to enable the setup of PVCs for use by IP."

::= { ipoaObjects 8 }

## ipoaConfigPvcEntry OBJECT-TYPE

SYNTAX IpoaConfigPvcEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"Defines a single PVC that exists at this host for use by IP."

```

INDEX      { ipoaConfigPvcIfIndex,
              ipoaConfigPvcVpi,
              ipoaConfigPvcVci
            }
 ::= { ipoaConfigPvcTable 1 }

IpoaConfigPvcEntry ::= SEQUENCE {
    ipoaConfigPvcIfIndex      InterfaceIndex,
    ipoaConfigPvcVpi          IpoaVpiInteger,
    ipoaConfigPvcVci          IpoaVciInteger,
    ipoaConfigPvcDefaultMtu   Integer32,
    ipoaConfigPvcRowStatus    RowStatus }

ipoaConfigPvcIfIndex OBJECT-TYPE
    SYNTAX      InterfaceIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The ifIndex of the ATM Interface that this PVC
         is associated with."
    ::= { ipoaConfigPvcEntry 1 }

ipoaConfigPvcVpi OBJECT-TYPE
    SYNTAX      IpoaVpiInteger
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The VPI value for the Virtual Circuit."
    ::= { ipoaConfigPvcEntry 2 }

ipoaConfigPvcVci OBJECT-TYPE
    SYNTAX      IpoaVciInteger
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The VCI value for the Virtual Circuit."
    ::= { ipoaConfigPvcEntry 3 }

ipoaConfigPvcDefaultMtu OBJECT-TYPE
    SYNTAX      Integer32 (0..65535)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Classical IP and ARP over ATM allows use of
         other MTU values for PVCs but considers how a
         value other than 9180 could be selected to be out
         of scope.  ipoaConfigPvcDefaultMtu can be used to
         configure the MTU to be used for the PVC."

```

Both ends MUST have the same value configured."  
 DEFVAL { 9180 }  
 ::= { ipoaConfigPvcEntry 4 }

#### ipoaConfigPvcRowStatus OBJECT-TYPE

SYNTAX RowStatus  
 MAX-ACCESS read-create  
 STATUS current  
 DESCRIPTION

"This object allows rows to be created and deleted in the ipoaConfigPvcTable. Creation of an entry in this table should eventually result in the creation of an ipNetToMediaEntry and a corresponding ipoaVcEntry after InATMARP has determined the destination address of the remote system that the PVC is connected to. Setting this object to destroy(6) should remove the corresponding ipNetToMediaTable and ipoaVcTable entries."

#### REFERENCE

"RFC 1903, 'Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2).'

::= { ipoaConfigPvcEntry 5 }

#### -- Notifications

ipoaTrapPrefix OBJECT IDENTIFIER ::= { ipoaNotifications 0 }

#### ipoaMtuExceeded NOTIFICATION-TYPE

OBJECTS {  
     ipoaVcNegotiatedMtu  
 }

STATUS current

#### DESCRIPTION

"A frame was received that exceeds the negotiated MTU size. The VPI and VCI of the VC for which this condition was detected can be determined from the index values for ipoaVcNegotiatedMtu. In addition, the ifIndex and IP Address can be determined as well (refer to the ipoaVcTable)."

::= { ipoaTrapPrefix 1 }

#### ipoaDuplicateIpAddress NOTIFICATION-TYPE

OBJECTS {  
     ipNetToMediaIfIndex,  
     ipNetToMediaNetAddress,  
     ipNetToMediaPhysAddress,  
     ipNetToMediaPhysAddress

```

    }
    STATUS current
    DESCRIPTION
        "The ATMARP Server has detected more than one ATM end
        point attempting to associate the same IP address with
        different ATM addresses."
    ::= { ipoaTrapPrefix 2 }

ipoaLisCreate NOTIFICATION-TYPE
    OBJECTS {
        ipoaLisSubnetAddr
    }
    STATUS current
    DESCRIPTION
        "Generation of this trap occurs when an ipoaLisEntry is
        created while the ipoaLisTrapEnable.0 object has the
        value enabled(1)."
```

```

    ::= { ipoaTrapPrefix 3 }

ipoaLisDelete NOTIFICATION-TYPE
    OBJECTS {
        ipoaLisSubnetAddr
    }
    STATUS current
    DESCRIPTION
        "Generation of this trap occurs when an ipoaLisEntry is
        deleted while the ipoaLisTrapEnable.0 object has the
        value enabled(1)."
```

```

    ::= { ipoaTrapPrefix 4 }

-- Conformance Definitions

ipoaGroups          OBJECT IDENTIFIER ::= { ipoaConformance 1 }

ipoaCompliances OBJECT IDENTIFIER ::= { ipoaConformance 2 }

-- compliance statements

ipoaCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
        "The compliance statement for agents that support the
        IPOA-MIB."
    MODULE -- this module
        MANDATORY-GROUPS { ipoaGeneralGroup,
                            ipoaBasicNotificationsGroup
                            }
        GROUP ipoaClientGroup

```



## DESCRIPTION

"This group is mandatory for all hosts where IP over ATM client support is present."

## GROUP ipoaSrvrGroup

## DESCRIPTION

"This group is mandatory for all hosts where ATMARP Servers are present."

## GROUP ipoaSrvrNotificationsGroup

## DESCRIPTION

"This group is mandatory for all hosts where ATMARP Servers are present."

## GROUP ipoaLisNotificationsGroup

## DESCRIPTION

"This group is mandatory for all hosts where ATMARP client only support is present and ipoaLisTrapEnable is allowed to be set to enabled(1)."

## GROUP ipoaLisTableGroup

## DESCRIPTION

"This group is mandatory for all entities which support IP over ATM SVCs. Support of objects in this group by IP over ATM clients which only support IP over ATM PVCs is optional."

## OBJECT ipoaLisDefaultMtu

MIN-ACCESS read-only

## DESCRIPTION

"The agent is not required to allow the user to change the default MTU from the value 9180."

The agent is not required to support a SET operation to this object in the absence of adequate security."

## OBJECT ipoaLisDefaultEncapsType

MIN-ACCESS read-only

## DESCRIPTION

"The agent is not required to allow the user to specify the default encapsulation type for the LIS."

The agent is not required to support a SET operation to this object in the absence of adequate security."

## OBJECT ipoaLisInactivityTimer

MIN-ACCESS read-only

## DESCRIPTION

"The agent is not required to support a SET operation to this object in the absence of adequate security."

OBJECT ipoaLisMinHoldingTime

MIN-ACCESS read-only

DESCRIPTION

"The agent is not required to support a SET operation to this object in the absence of adequate security."

OBJECT ipoaLisQDepth

MIN-ACCESS read-only

DESCRIPTION

"The agent is not required to support a SET operation to this object in the absence of adequate security."

OBJECT ipoaLisMaxCalls

MIN-ACCESS read-only

DESCRIPTION

"The agent is not required to support a SET operation to this object in the absence of adequate security."

OBJECT ipoaLisCacheEntryAge

MIN-ACCESS read-only

DESCRIPTION

"The agent is not required to support a SET operation to this object in the absence of adequate security."

OBJECT ipoaLisRetries

MIN-ACCESS read-only

DESCRIPTION

"The agent is not required to allow the user to change the default number of times an ATMARP request will be retried when no response is received from the default of 2.

The agent is not required to support a SET operation to this object in the absence of adequate security."

OBJECT ipoaLisTimeout

MIN-ACCESS read-only

DESCRIPTION

"The agent is not required to allow the user

to change the default retransmission time from the default of 10 seconds.

The agent is not required to support a SET operation to this object in the absence of adequate security."

OBJECT ipoaLisDefaultPeakCellRate

MIN-ACCESS read-only

DESCRIPTION

"Implementations that do not support IP over ATM SVC usage are not required to allow the user to specify a best effort default peak cell rate since typically the ipoaLisTable won't exist.

The agent is not required to support a SET operation to this object in the absence of adequate security."

OBJECT ipoaLisIfMappingRowStatus

SYNTAX INTEGER {  
                                active(1) -- subset of RowStatus  
                                }

MIN-ACCESS read-only

DESCRIPTION

"The agent is not required to support a SET operation to this object, and only one of the six enumerated values for the RowStatus textual convention need be supported, specifically: active(1)."

OBJECT ipoaArpClientAtmAddr

MIN-ACCESS read-only

DESCRIPTION

"The agent is not required to support a SET operation to this object in the absence of adequate security."

OBJECT ipoaArpSrvrLis

MIN-ACCESS read-only

DESCRIPTION

"The agent is not required to support a SET operation to this object in the absence of adequate security."

OBJECT ipoaArpRemoteSrvrAdminStatus

MIN-ACCESS read-only

## DESCRIPTION

"The agent is not required to support a SET operation to this object in the absence of adequate security. In this case the value of this object should be up(1) when a VC exists to the Remote ATMARP Server or otherwise down(2), and the agent should not allow a SET operation to this object."

OBJECT ipoaConfigPvcDefaultMtu

MIN-ACCESS read-only

## DESCRIPTION

"The agent is not required to support a SET operation to this object in the absence of adequate security."

OBJECT ipoaLisRowStatus

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 ipoaArpClientRowStatus

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 ipoaArpRemoteSrvrRowStatus

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 ipoaArpSrvrRowStatus
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 ipoaConfigPvcRowStatus
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 ipoaArpClientArpOutNaks

```

```

MIN-ACCESS not-accessible

```

```

DESCRIPTION

```

```

    "Classic IP and ARP over ATM does not require
    an ATMARP client to transmit an ATMARP_NAK
    upon receipt of an ATMARP request from another
    ATMARP client. This object should be
    implemented when an ATMARP client supports the
    transmission of ATMARP_NAKs."
```

```

 ::= { ipoaCompliances 1 }

```

```

-- units of conformance

```

```

ipoaGeneralGroup OBJECT-GROUP

```

```

    OBJECTS {
        ipoaVcType,
        ipoaVcNegotiatedEncapsType,
        ipoaVcNegotiatedMtu,
        ipoaConfigPvcDefaultMtu,
        ipoaConfigPvcRowStatus
    }

```

```

STATUS current

```

```

DESCRIPTION

```

```

    "This group is mandatory for all IP over ATM entities."
```

```

 ::= { ipoaGroups 1 }

```

## ipoaClientGroup OBJECT-GROUP

```
OBJECTS {
    ipoaArpClientAtmAddr,
    ipoaArpClientSrvrInUse,
    ipoaArpClientInArpInReqs,
    ipoaArpClientInArpOutReqs,
    ipoaArpClientInArpInReplies,
    ipoaArpClientInArpOutReplies,
    ipoaArpClientInArpInvalidInReqs,
    ipoaArpClientInArpInvalidOutReqs,
    ipoaArpClientArpInReqs,
    ipoaArpClientArpOutReqs,
    ipoaArpClientArpInReplies,
    ipoaArpClientArpOutReplies,
    ipoaArpClientArpInNaks,
    ipoaArpClientArpOutNaks,
    ipoaArpClientArpUnknownOps,
    ipoaArpClientArpNoSrvrResps,
    ipoaArpClientRowStatus
}
```

STATUS current

## DESCRIPTION

"This group is mandatory for all hosts where an IP  
over ATM client is present."

::= { ipoaGroups 2 }

## ipoaSrvrGroup OBJECT-GROUP

```
OBJECTS {
    ipoaArpSrvrLis,
    ipoaArpSrvrInArpInReqs,
    ipoaArpSrvrInArpOutReqs,
    ipoaArpSrvrInArpInReplies,
    ipoaArpSrvrInArpOutReplies,
    ipoaArpSrvrInArpInvalidInReqs,
    ipoaArpSrvrInArpInvalidOutReqs,
    ipoaArpSrvrArpInReqs,
    ipoaArpSrvrArpOutReplies,
    ipoaArpSrvrArpOutNaks,
    ipoaArpSrvrArpDupIpAddrs,
    ipoaArpSrvrArpUnknownOps,
    ipoaArpSrvrRowStatus
}
```

STATUS current

## DESCRIPTION

"This group is mandatory for all hosts where ATMARP  
Servers are present."

::= { ipoaGroups 3 }

```
ipoaBasicNotificationsGroup NOTIFICATION-GROUP
  NOTIFICATIONS {
    ipoaMtuExceeded
  }
  STATUS          current
  DESCRIPTION
    "The notification which an IP over ATM entity
    is required to implement."
  ::= { ipoaGroups 4 }

ipoaSrvrNotificationsGroup NOTIFICATION-GROUP
  NOTIFICATIONS {
    ipoaDuplicateIpAddress
  }
  STATUS          current
  DESCRIPTION
    "The notification which an IP over ATM ATMARP
    Server is required to implement."
  ::= { ipoaGroups 5 }

ipoaLisNotificationsGroup NOTIFICATION-GROUP
  NOTIFICATIONS {
    ipoaLisCreate,
    ipoaLisDelete
  }
  STATUS          current
  DESCRIPTION
    "The LIS-related notifications which are required
    to be implemented by an IP over ATM ATMARP server,
    as well as by any IP over ATM client which allows
    ipoaLisTrapEnable to be set to enabled(1)."
  ::= { ipoaGroups 6 }

ipoaLisTableGroup OBJECT-GROUP
  OBJECTS {
    ipoaLisTrapEnable,
    ipoaLisSubnetAddr,
    ipoaLisDefaultMtu,
    ipoaLisDefaultEncapsType,
    ipoaLisInactivityTimer,
    ipoaLisMinHoldingTime,
    ipoaLisQDepth,
    ipoaLisMaxCalls,
    ipoaLisCacheEntryAge,
    ipoaLisRetries,
    ipoaLisTimeout,
    ipoaLisDefaultPeakCellRate,
    ipoaLisActiveVcs,
```

```
    ipoaLisRowStatus,
    ipoaLisIfMappingRowStatus,
    ipoaArpRemoteSrvrRowStatus,
    ipoaArpRemoteSrvrIpAddr,
    ipoaArpRemoteSrvrAdminStatus,
    ipoaArpRemoteSrvrOperStatus
}
STATUS    current
DESCRIPTION
    "This group is mandatory for all entities which
    support IP over ATM SVCs. Support of objects in
    this group by IP over ATM clients which only
    support IP over ATM PVCs is optional."
 ::= { ipoaGroups 7 }
```

END

## 5. Security Considerations

Certain management information defined in this MIB MAY be considered sensitive in some network environments. Therefore, authentication of received SNMP requests and controlled access to management information SHOULD be employed in such environments. The method for this authentication is a function of the SNMP Administrative Framework, and has not been expanded by this MIB.

Several objects in this MIB allow write access or provide for row creation. Allowing this support in a non-secure environment can have a negative effect on network operations. It is RECOMMENDED that implementers seriously consider whether set operations or row creation be allowed without providing, at a minimum, authentication of request origin. It is RECOMMENDED that without such support that the following objects be implemented as read-only:

- o ipoaLisDefaultMtu
- o ipoaLisDefaultEncapsType
- o ipoaLisInactivityTimer
- o ipoaLisMinHoldingTime
- o ipoaLisQDepth
- o ipoaLisMaxCalls
- o ipoaLisCacheEntryAge
- o ipoaLisRetries
- o ipoaLisTimeout
- o ipoaLisDefaultPeakCellRate
- o ipoaArpClientAtmAddr
- o ipoaArpSrvrLis



- o ipoaArpRemoteSrvrAdminStatus, show status as being either up(1) when a VC exists to the Remote ATMARP Server or otherwise down(2). Don't allow set support.
- ipoaArpRemoteSrvrOperStatus would have the same value as ipoaArpRemoteSrvrAdminStatus.
- o ipoaConfigPvcDefaultMtu
- o ipoaLisRowStatus
- o ipoaArpClientRowStatus
- o ipoaArpRemoteSrvrRowStatus
- o ipoaArpSrvrRowStatus
- o ipoaConfigPvcRowStatus
- o ipoaLisIfMappingRowStatus

## 6. Intellectual Property

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

## 7. Acknowledgments

This document is a product of the Internetworking Over NBMA Working Group. The authors of this document would like to recognize Keith McCloghrie from Cisco Systems for his support as our mentor from the Network Management Area.

## 8. 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 1902, January 1996.
- [2] McCloghrie, K., and F. Kastenholz, "The Interfaces Group MIB using SMIV2", RFC 2233, November 1997.
- [3] Laubach M., and J. Halpern, "Classical IP and ARP over ATM", RFC 2225, April 1998.
- [4] Ahmed, M., and K. Tesink, "Definitions of Managed Objects for ATM Management Version 8.0 using SMIV2", RFC 1695, August 1994.
- [5] McCloghrie, K., and M. Rose, Editors, "Management Information Base for Network Management of TCP/IP-based internets: MIB-II", STD 17, RFC 1213, March 1991.
- [6] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1903, January 1996.
- [7] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1905, January 1996.
- [8] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1904, January 1996.
- [9] McCloghrie K., "Management Information Base for the Internet Protocol using SMIV2", RFC 2011, November 1996.
- [10] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

- [11] Perez, M., Liaw, F., Mankin, A., Hoffman, E., Grossman, D. and A. Malis, "ATM Signaling Support for IP over ATM", RFC 1755, February 1995.
- [12] Case, J., McCloghrie, K., Rose, M., and Waldbusser, S., "Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1906, January 1996.
- [13] Case, J., McCloghrie, K., Rose, M., and Waldbusser, S., "Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1907, January 1996.
- [14] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Coexistence between Version 1 and Version 2 of the Internet-standard Network Management Framework", RFC 1908, January 1996.

## 9. Authors' Addresses

Maria N. Greene  
Xedia Corp.  
119 Russell Dr.  
Littleton, MA 01460  
EMail: maria@xedia.com

James Luciani  
Bay Networks, Inc.  
3 Federal St., BL3-04  
Billerica, MA 01821, USA  
Phone: +1-508-439-4734  
EMail: luciani@baynetworks.com

Kenneth D. White  
Dept. G80/Bldg 503  
IBM Corporation  
Research Triangle Park, NC 27709, USA  
EMail: kennethw@vnet.ibm.com

Ted T.I. Kuo  
Bay Networks, Inc.  
4401 Great America Parkway  
Santa Clara, CA 95052-8185  
Phone: +1-408-495-7319  
Fax: +1-408-495-1905  
EMail: ted\_kuo@Baynetworks.com

## 10. Full Copyright Statement

Copyright (C) The Internet Society (1998). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

