

Definitions of Tests
for ATM Management

November 28, 1999

Internet-Draft <[draft-ietf-atommib-test-06.txt](#)>

Michael Noto (editor)
3Com Corporation
mike_noto@3com.com

Kaj Tesink (editor)
Telcordia Technologies
kaj@research.telcordia.com

1. Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

Copyright Notice

Copyright (C) The Internet Society (1999). All Rights Reserved.

2. Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes objects used for managing ATM-based interfaces, devices, networks and services in addition to those defined in the ATM MIB [[RFC2515](#)], to provide support for the management of on-demand ATM Loopback Tests.

3. The SNMP Network Management Framework

The SNMP Management Framework presently consists of five major components:

- o An overall architecture, described in [RFC 2571](#) [[RFC2571](#)].
- o Mechanisms for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIV1 and described in STD 16, [RFC 1155](#) [[RFC1155](#)], STD 16, [RFC 1212](#) [[RFC1212](#)] and [RFC 1215](#) [[RFC1215](#)]. The second version, called SMIV2, is described in STD 58, [RFC 2578](#) [[RFC2578](#)], [RFC 2579](#) [[RFC2579](#)] and [RFC 2580](#) [[RFC2580](#)].
- o Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in STD 15, [RFC 1157](#) [[RFC1157](#)]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in [RFC 1901](#) [[RFC1901](#)] and [RFC 1906](#) [[RFC1906](#)]. The third version of the message protocol is called SNMPv3 and described in [RFC 1906](#) [[RFC1906](#)], [RFC 2572](#) [[RFC2572](#)] and [RFC 2574](#) [[RFC2574](#)].

Expires May 28, 2000

[Page 2]

- o Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in STD 15, [RFC 1157](#) [[RFC1157](#)]. A second set of protocol operations and associated PDU formats is described in [RFC 1905](#) [[RFC1905](#)].
- o A set of fundamental applications described in [RFC 2573](#) [[RFC2573](#)] and the view-based access control mechanism described in [RFC 2575](#) [[RFC2575](#)].

A more detailed introduction to the current SNMP Management Framework can be found in [RFC 2570](#) [[RFC2570](#)].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the mechanisms defined in the SMI.

This memo specifies a MIB module that is compliant to the SMIV2. A MIB conforming to the SMIV1 can be produced through the appropriate translations. The resulting translated MIB must be semantically equivalent, except where objects or events are omitted because no translation is possible (use of Counter64). Some machine readable information in SMIV2 will be converted into textual descriptions in SMIV1 during the translation process. However, this loss of machine readable information is not considered to change the semantics of the MIB.

Expires May 28, 2000

[Page 3]

4. Overview

The purpose of this memo is to provide additional capabilities, not found in the ATM MIB [[RFC2515](#)], which are needed to manage ATM interfaces. This memo addresses ATM Testing Support and must be used in conjunction with the System/Interface Test MIB [[TestMIB](#)].

4.1. Background

In addition to the MIB module defined in this memo, other MIB modules are necessary to manage ATM interfaces, links and cross-connects. Examples include MIB II for general system [[RFC1213](#)] and interface management [[RFC2233](#)], the DS3 or SONET/SDH MIBs for management of SONET/SDH and DS3 physical interfaces, and, as appropriate, MIB modules for applications that make use of ATM, such as SMDS and LAN Emulation. These MIB modules are outside the scope of this specification.

This MIB module requires the use of the ATM MIB module defined in [[RFC2515](#)] and the System/Interface Test MIB module [[TestMIB](#)].

This memo defines extensions to the ATM MIB in order to support ATM Loopback Tests. An ATM Loopback Test provides the ability to send out a loopback OAM (Operations and Maintenance) cell to verify the existence of connectivity for a particular connection.

Expires May 28, 2000

[Page 4]

4.2. Terminology

The following terms are defined here and used throughout this MIB:

- Virtual Path Link (VPL)
- Virtual Path Connection (VPC)
- Virtual Path Segment (VP Segment)
- Virtual Channel Link (VCL)
- Virtual Channel Connection (VCC)
- Virtual Channel Segment (VC Segment).

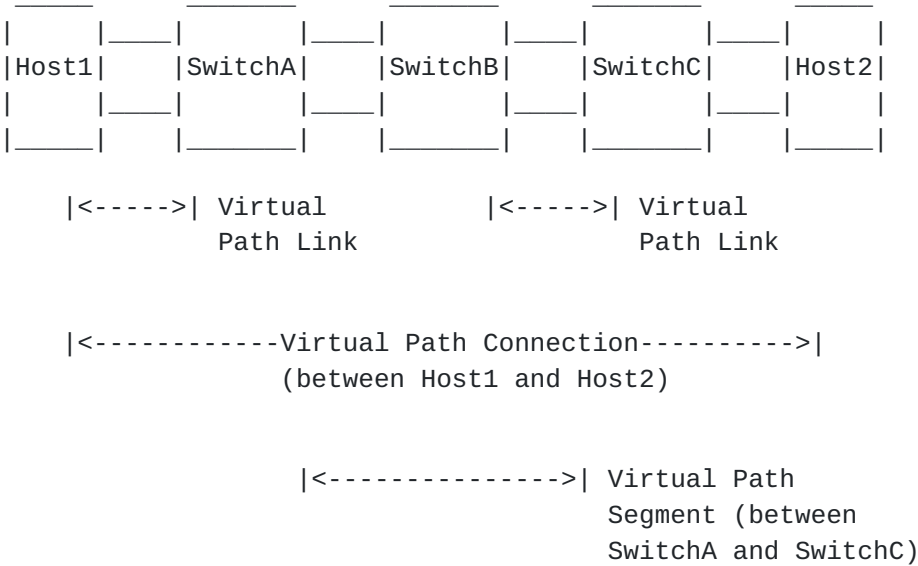


Figure 1: Examples of Virtual Path Links, Virtual Path Connection, and Virtual Path Segment

Expires May 28, 2000

[Page 5]

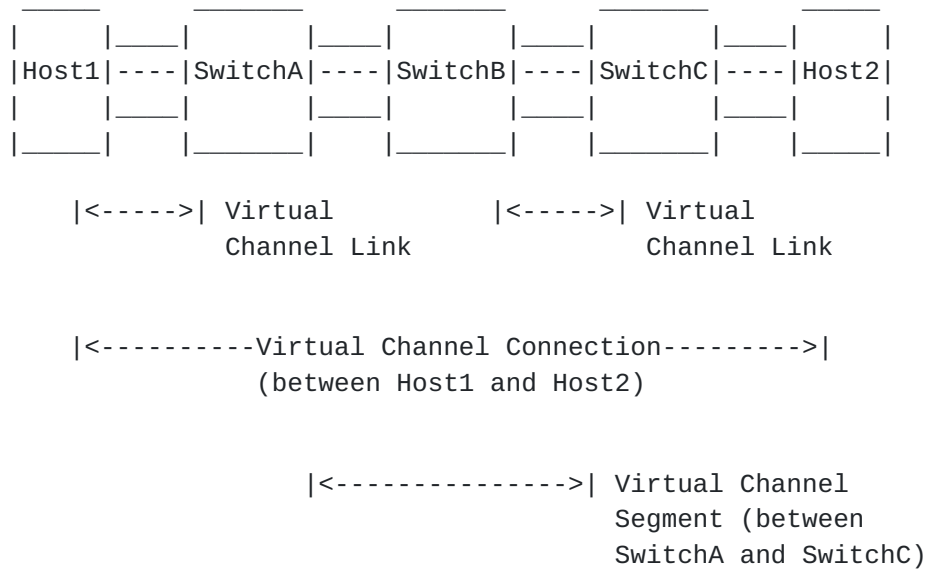


Figure 2: Examples of Virtual Channel Links, Virtual Channel Connection, and Virtual Channel Segment

[4.3.](#) Supported Functions

The managed ATM objects are organized as follows:

- (1) ATM Loopback Testing
- (2) ATM End-Point Tables

[4.3.1.](#) ATM Loopback Testing

The loopback test provides the ability to send out a loopback OAM cell to verify the existence of connectivity for a particular connection. Loopback tests can be performed on either an entire connection (i.e., an end-to-end test), a segment of the connection (i.e., a segment test), a portion of a segment (i.e., a loopback location identifier test), or the network portion of a connection (i.e., a service internal test).

The loopback test makes use of the Test Table defined in [[TestMIB](#)]. For a given interface, a loopback test can be invoked by obtaining ownership of a test and then by setting the value of testType equal to one of the ATM Loopback Test Types defined in [Section 5](#). See procedures in [[TestMIB](#)] for

Expires May 28, 2000

[Page 6]

using the Test Table. After invoking a loopback test, the object testResult can be read to determine the outcome of the loopback test (e.g., 'success(2)' if the loopback cell made it back to the originator of the test or 'failed(7)' if the loopback cell did not make it back).

The following types of loopback tests are defined:

- End-to-end Loopback Test
- Segment Loopback Test
- Loopback Test Using Loopback Location Identifier
- Network Loopback Test.

1) End-to-end Loopback Test

The end-to-end loopback (LB) is self-explanatory. For a VP test, the cell is sent on the given VP, via VCI=4 specified in [UNI3.1]. For a VC test, the LB cell is sent on the VC under test, with the PTI (Payload Type Indicator) set to 5 as specified in [UNI3.1]. Figure 3 illustrates the end-to-end loopback test.

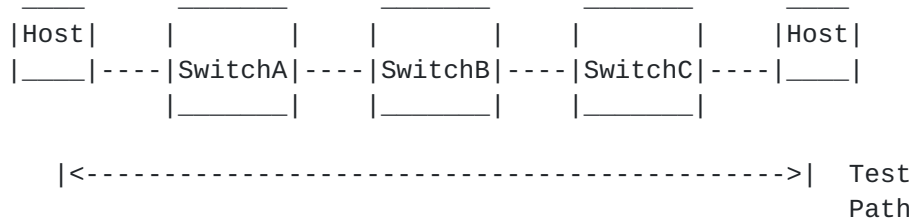


Figure 3: End-to-end Loopback Test

See [Section 5](#) for more details on how to use the End-to-end Loopback Test.

2) Segment Loopback Test

The segment LB test is explained in ITU-T I.610[I.610]. For a VP segment test, the LB cell is sent on the VP under test via VCI=3 as specified in [UNI3.1], and the Loopback Location ID field is set to all 1's. For a VC segment test, the LB cell is sent on the VC under test, with the PTI set to 4 as

Expires May 28, 2000

[Page 7]

specified in [[UNI3.1](#)], and the Loopback Location ID field is set to all 1's.

This test involves a LB cell being inserted at a pre-defined segment end-point, and looped back at the corresponding segment end-point encountered. The pair of segment end-points define a segment (which is used for the segment loopback test). A VP/VC connection can have multiple segments, but multiple segments cannot overlap.

A UNI interface is by definition defined as a segment end-point (hence a UNI would be considered a segment). A segment can also define:

- a B-ICI
- a public carrier's 'piece' of the connection
- a private network's 'piece' of the connection.

In order to support this functionality, the VP/VC link termination needs to be able to be defined as a segment. This can be done using either the atmVplSegmentEndPoint or atmVclSegmentEndPoint object depending on whether it is for a VPC or VCC. A segment loopback test is illustrated in Figure 4.

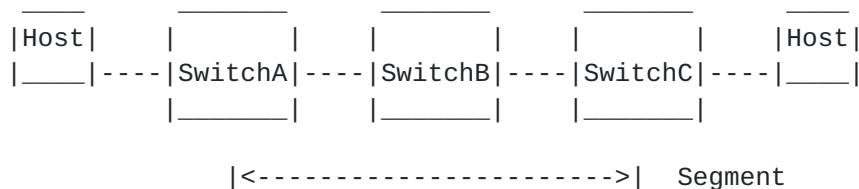


Figure 4: Segment Loopback Test

[Section 5](#) describes the use of the ATM Segment Loopback Tests.

3) Loopback Test Using Loopback Location Identifier

This loopback test is a special type of 2) where the Loopback Location ID field is set to a value that corresponds to a specific node in a given network (Note that the format of this field is not standardized, that is, the value is significant only within an administrative domain). In this case, the device initiating the LB test inserts the appropriate Loop

Expires May 28, 2000

[Page 8]

Back Location ID. When the LB cell reaches the corresponding device, that device recognizes the Loopback Location ID as its own, and loops it back. This test is useful for performing fault sectionalization without having to provision segment end-points. An additional object, the `atmIfLogicalLoopbackId`, is defined to determine the loopback point. Figure 5 shows a loopback test using a location identifier. Note that the loopback test using location identifier can be used to perform a loopback test over a portion of a defined segment. See Figure 5.

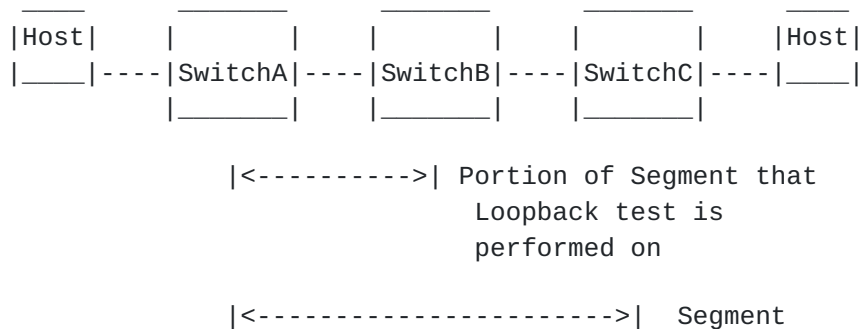


Figure 5: Loopback Test Using Location Identifier

See [Section 5](#) for more details.

4) Network Loopback Test

This is a loopback test that the manager requests an agent in a network to perform over the internal portion of a designated connection. The Network then initiates the internal network loopback test by inserting an OAM loopback cell at one of the end-points of the internal network portion of the connection. When the loopback cell reaches the other end-point of the internal Network, the cell is looped back. This test is useful for verifying connectivity through a particular network. Figure 6 illustrates the Network loopback test.

Expires May 28, 2000

[Page 9]

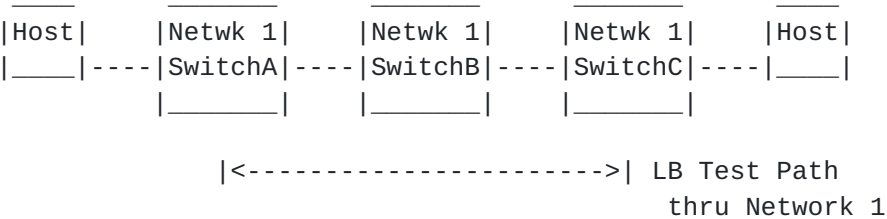


Figure 6: Network Loopback Test

See [Section 5](#) for more details.

4.3.2. ATM End-Point Tables

There are two ATM End-point tables: the ATM VP End-point Table and the ATM VC End-point Table. The ATM VP End-point Table augments the atmVplTable and defines the atmVplEndptSegmentEndPoint object to represent whether or not a specified VPL is a segment end-point. Similarly for Virtual Channels, the ATM VC End-point Table and the atmVclEndptSegmentEndPoint object are used to represent whether or not a specified VCL is a segment end-point.

5. Definitions

```
ATMTEST-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, OBJECT-IDENTITY,
    OBJECT-TYPE, mib-2
        FROM SNMPv2-SMI
    MODULE-COMPLIANCE, OBJECT-GROUP
        FROM SNMPv2-CONF
    atmVplEntry, atmVclEntry, atmInterfaceConfEntry
        FROM ATM-MIB;

atmTESTMIB MODULE-IDENTITY
    LAST-UPDATED "9911281200Z"
    ORGANIZATION "IETF AToMMIB Working Group"
    CONTACT-INFO
        "
            Michael Noto
            Postal: 3Com Corporation
                  5400 Bayfront Plaza
                  Santa Clara, CA 95052
                  USA
            Tel:    +1 408 326 2218
            E-mail: mike_noto@3com.com

            Kaj Tesink
            Postal: Telcordia Technologies
                  331 Newman Springs Road
                  Red Bank, NJ 07701
                  USA
            Tel:    +1 732 758 5254
            Fax:    +1 732 758 2269
            E-mail: kaj@research.telcordia.com"
    DESCRIPTION
        "This MIB Module provides
        ATM Loopback Tests and supporting objects
        that must be supported by ATM devices
        providing ATM Loopback Tests."
    REVISION      "9911281200Z"
    DESCRIPTION
        "Initial version, published as RFCxxxx"
    ::= { mib-2 XX }

-- ***** NOTE TO THE RFC EDITOR / IANA *****
```

Expires May 28, 2000

[Page 11]

```
-- * Before this module is put on the standards track
-- * fill out RFCxxxx with the RFC number of this document
-- * assign XX by IANA.
-- * remove this notice from the MIB
```

```
atmTESTMIBObjects OBJECT IDENTIFIER ::= {atmTESTMIB 1}
```

```
-- This ATMTEST-MIB Module consists of the following:
--      (1) ATM Loopback Testing
--      (2) ATM End-Point Tables
```

```
-- *****
-- (1) ATM Loopback Testing

-- This concerns information for interfaces
-- supporting ATM Loopback Tests and includes:
-- 1. ATM Loopback Objects
-- 2. List of ATM Loopback Test Types

atmLoopbackTestGroup OBJECT IDENTIFIER ::= {
    atmTESTMIBObjects 1}

-- 1. ATM Loopback Objects
-- The following objects are defined for use in
-- performing ATM Loopback Tests.

atmInterfaceLLIDTable      OBJECT-TYPE
    SYNTAX      SEQUENCE OF AtmInterfaceLLIDEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "A table of Logical Loopback Identifiers (LLID)
        for ATM interfaces"
    ::= { atmLoopbackTestGroup 1 }

atmInterfaceLLIDEntry OBJECT-TYPE
    SYNTAX      AtmInterfaceLLIDEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "A conceptual row in the atmInterfaceLLIDTable."
    AUGMENTS    { atmInterfaceConfEntry }
    ::= { atmInterfaceLLIDTable 1 }

AtmInterfaceLLIDEntry ::= SEQUENCE {
    atmIfLogicalLoopbackId  OCTET STRING
}

atmIfLogicalLoopbackId OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE (0|16))
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "This identifier is used to identify this local ATM
```

Expires May 28, 2000

[Page 13]

interface for loopback. The value of this object can be used by other ATM devices to identify this local ATM interface as the interface that is being requested to loopback OAM Loopback cells. The default for this field is the zero length string which signifies that loopback at this interface is disabled. To enable loopback set the value to a 16 byte fixed length octet string. Non-zero length identifiers of less than 16 octets signify that loopback checking is enabled and are left justified, and padded with all 0's."

DEFVAL {'H' }

REFERENCE

"ITU-T Recommendation I.610, Broadband Integrated Service Digital Network (B-ISDN) Operation and Maintenance Principles and Functions, July 1995"

::= { atmInterfaceLLIDEntry 1 }

-- 2. List of ATM Loopback Test Types

-- The following loopback test types are defined:

```
-- atmLoopbackVpE2e
-- atmLoopbackVcE2e
-- atmLoopbackVpSegment
-- atmLoopbackVcSegment
-- atmLoopbackVpLocationID
-- atmLoopbackVcLocationID
-- atmLoopbackVpServiceInternal
-- atmLoopbackVcServiceInternal
```

```
atmLoopbackTestTypes OBJECT IDENTIFIER ::= {
    atmLoopbackTestGroup 4 }
```

atmLoopbackVpE2e OBJECT-IDENTITY

STATUS current

DESCRIPTION

"This is an end-to-end loopback test performed on a designated VP (Virtual Path). To perform this test an end-to-end loopback OAM cell is inserted at one of the end-points of the designated VP connection (e.g., at a host) via VCI=4 (the VCI value for VP OAM end-to-end cells), travels to the other end-point of the VP

Expires May 28, 2000

[Page 14]

connection, and then loops back to the originating end-point on the designated VP. Success is achieved if the loopback OAM cell returns to the originating end-point within 5 seconds, otherwise, the test fails.

The manager-station performs a loopback test by making use of the testTable defined in [TestMIB]. In order to run this test the object testType in the testTable shall be set to atmLoopbackVpE2e, and the object testTarget points to the row in the atmVplTable in [TestMIB] corresponding to the VP designated for the test.

Before starting a test, a manager-station must first obtain 'ownership' of the entry in the testTable for the interface to be tested (follow procedure defined in [TestMIB]). Once the manager-station obtains ownership, a loopback test for a given interface can be invoked by first setting up parameters necessary for the loopback test (e.g., set the testTarget), and then setting the value of testType in the testTable equal to 'atmLoopbackVpE2e'. The testRowStatus is used to invoke the atmLoopbackVpE2e test on the VP with the VPI corresponding to the testTarget.

After invoking a loopback test, wait for the test completion by polling for the object testResult. A value of 'InProgress(3)' will result if the test is still in progress. Once the test is completed, the object testResult will have a value of 'success(2)' if the loopback OAM cell returned to the originator of the test within 5 seconds, if not, a value of 'failed(7)' will result. If the ATM system does not support this type of loopback test, then a value of 'notSupported(4)' will be provided. Other possible values for the testResult object are 'unableToRun(5)' and 'aborted(6)'."

```
::= { atmLoopbackTestTypes 1 }
```

atmLoopbackVce2e OBJECT-IDENTITY

STATUS current

DESCRIPTION

"This is an end-to-end loopback test performed on a designated VC (Virtual Channel). To perform this test

Expires May 28, 2000

[Page 15]

an end-to-end loopback OAM cell is inserted at one of the end-points of the designated VC connection (e.g., at a host) via PTI=5 (the PTI value used for VC OAM end-to-end cells), travels to the other end-point of the VC connection, and then loops back to the originating end-point on the designated VC. Success is achieved if the loopback OAM cell returns to the originating end-point within 5 seconds, otherwise, the test fails.

The manager-station performs a loopback test by making use of the testTable defined in [TestMIB]. In order to run this test the object testType in the testTable shall be set to atmLoopbackVcE2e, and the object testTarget points to the row in the atmVclTable in [RFC2515] corresponding to the VC designated for the test.

Before starting a test, a manager-station must first obtain 'ownership' of the entry in the testTable for the interface to be tested (follow procedure defined in [TestMIB]). Once the manager-station obtains ownership, a loopback test for a given interface can be invoked by first setting up parameters necessary for the loopback test (e.g., set the testTarget), and then setting the value of testType in the testTable equal to 'atmLoopbackVcE2e'. The testRowStatus is used to invoke the atmLoopbackVcE2e test on the VC with the VPI/VCI corresponding to the testTarget.

After invoking a loopback test, wait for the test completion by polling for the object testResult. A value of 'InProgress(3)' will result if the test is still in progress. Once the test is completed, the object testResult will have a value of 'success(2)' if the loopback OAM cell returned to the originator of the test within 5 seconds, if not, a value of 'failed(7)' will result. If the ATM system does not support this type of loopback test, then a value of 'notSupported(4)' will be provided. Other possible values for the testResult object are 'unableToRun(5)' and 'aborted(6)'."

::= { atmLoopbackTestTypes 2 }

Expires May 28, 2000

[Page 16]

atmLoopbackVpSegment OBJECT-IDENTITY

STATUS current

DESCRIPTION

"This is a loopback test performed on a designated segment of a VP (Virtual Path). To perform this test a segment OAM cell is inserted at one of the segment end-points of the designated VP connection (e.g., at a host) via VCI=3 (the VCI used for VP OAM segment cells), travels across the segment on the designated VP to the device pre-configured as the corresponding segment end-point, and then loops back to the originating segment end-point on the designated VP. Success is achieved if the loopback OAM cell returns to the originating end-point within 5 seconds, otherwise, the test fails.

In order to use the atmLoopbackVpSegment test, a segment must be defined by setting up segment end-points using the atmVplEndptSegmentEndPoint object from the atmVplEndptTable. The atmVplEndptSegmentEndPoint is set to 'isaVpSegmentEndPoint(1)' for each segment end-point. Note that this object is by default set to 'isaVpSegmentEndPoint(1)' if the atmVplTable supports one end of a UNI. In such a case, a UNI VP loopback test would be achieved when the atmLoopbackVpSegment test was initiated over the UNI.

The manager-station performs a loopback test by making use of the testTable defined in [[TestMIB](#)]. In order to run this test the object testType in the testTable shall be set to atmLoopbackVpE2e, and the object testTarget points to the row in the atmVplTable in [[RFC2515](#)] corresponding to the VP designated for the test.

Before starting a test, a manager-station must first obtain 'ownership' of the entry in the testTable for the interface to be tested (follow procedure defined in [[TestMIB](#)]). Once the manager-station obtains ownership, a loopback test for a given interface can be invoked by first setting up parameters necessary for the loopback test (e.g., set the testTarget), and then setting the value of testType in the testTable equal to 'atmLoopbackVpSegment'. The testRowStatus is used to invoke the atmLoopbackVpSegment test on the VP with the

Expires May 28, 2000

[Page 17]

VPI corresponding to the testTarget.

After invoking a loopback test, wait for the test completion by polling for the object testResult. A value of 'InProgress(3)' will result if the test is still in progress. Once the test is completed, the object testResult will have a value of 'success(2)' if the loopback OAM cell returned to the originator of the test within 5 seconds, if not, a value of 'failed(7)' will result. If the ATM system does not support this type of loopback test, then a value of 'notSupported(4)' will be provided. Other possible values for the testResult object are 'unableToRun(5)' and 'aborted(6)'."

::= { atmLoopbackTestTypes 3 }

atmLoopbackVcSegment OBJECT-IDENTITY

STATUS current

DESCRIPTION

"This is a loopback test performed on a designated segment of a VC (Virtual Channel). To perform this test a segment OAM cell is inserted at one of the segment end-points of the designated VC connection (e.g., at a host) via PTI=4 (the PTI value used for VC OAM segment cells), travels across the segment on the designated VC to the device pre-configured as the corresponding segment end-point, and then loops back to the originating segment end-point on the designated VC. Success is achieved if the loopback OAM cell returns to the originating end-point within 5 seconds, otherwise, the test fails.

In order to use the atmLoopbackVcSegment test, a segment must be defined by setting up segment end-points using the atmVclEndptSegmentEndPoint object from the atmVclEndptTable. The atmVclEndptSegmentEndPoint is set to 'isaVcSegmentEndPoint(1)' for each segment end-point. Note that this object is by default set to 'isaVcSegmentEndPoint(1)' if the atmVclTable supports one end of a UNI. In such a case, a UNI VC loopback test would be achieved when the atmLoopbackVcSegment test was initiated over the UNI.

The manager-station performs a loopback test by making

Expires May 28, 2000

[Page 18]

use of the testTable defined in [[TestMIB](#)]. In order to run this test the object testType in the testTable shall be set to atmLoopbackVcE2e, and the object testTarget points to the row in the atmVclTable in [[RFC2515](#)] corresponding to the VC designated for the test.

Before starting a test, a manager-station must first obtain 'ownership' of the entry in the testTable for the interface to be tested (follow procedure defined in [[TestMIB](#)]). Once the manager-station obtains ownership, a loopback test for a given interface can be invoked by first setting up parameters necessary for the loopback test (e.g., set the testTarget), and then setting the value of testType in the testTable equal to 'atmLoopbackVcSegment'. The testRowStatus is used to invoke the atmLoopbackVcSegment test on the VC with the VPI/VCI corresponding to the testTarget.

After invoking a loopback test, wait for the test completion by polling for the object testResult. A value of 'InProgress(3)' will result if the test is still in progress. Once the test is completed, the object testResult will have a value of 'success(2)' if the loopback OAM cell returned to the originator of the test within 5 seconds, if not, a value of 'failed(7)' will result. If the ATM system does not support this type of loopback test, then a value of 'notSupported(4)' will be provided. Other possible values for the testResult object are 'unableToRun(5)' and 'aborted(6)'."

::= { atmLoopbackTestTypes 4 }

atmLoopbackVpLocationId OBJECT-IDENTITY

STATUS current

DESCRIPTION

"This is a loopback test performed on a portion of a designated VP segment. To perform this test a loopback OAM cell is inserted at a connection point of the designated VP connection (e.g., the end-point or a tandem point) with a value inserted in the Location Identifier ID field of the OAM cell that corresponds to the ATM device where the cell is to be looped back. The loopback cell then travels through the VP

Expires May 28, 2000

[Page 19]

connection until it reaches the designated ATM device, where it is looped back to the loopback cell insertion point on the designated VP. Success is achieved if the loopback OAM cell returns to the originating point of insertion within 5 seconds, otherwise, the test fails.

The manager-station performs a loopback test by making use of the testTable defined in [TestMIB]. In order to run this test the object testType in the testTable shall be set to atmLoopbackVpE2e. The object testTarget points to the row in the atmVpITable in [RFC2515] corresponding to the VP designated for the test. The object testMoreInfo contains the desired Loopback Location ID.

Before starting a test, a manager-station must first obtain 'ownership' of the entry in the testTable for the interface to be tested (follow procedure defined in [TestMIB]). Once the manager-station obtains ownership, a loopback test for a given interface can be invoked by first setting up parameters necessary for the loopback test (e.g., set testMoreInfo to AAAABBBBCCCCDDDD and set the testTarget), and then setting the value of testType in the testTable equal to 'atmLoopbackVpSegment'. The testRowStatus is used to invoke the atmLoopbackVpLocationId on the VP with the VPI corresponding to the testTarget and looped back at loopback location ID= AAAABBBBCCCCDDDD.

After invoking a loopback test, wait for the test completion by polling for the object testResult. A value of 'InProgress(3)' will result if the test is still in progress. Once the test is completed, the object testResult will have a value of 'success(2)' if the loopback OAM cell returned to the originator of the test within 5 seconds, if not, a value of 'failed(7)' will result. If the ATM system does not support this type of loopback test, then a value of 'notSupported(4)' will be provided. Other possible values for the testResult object are 'unableToRun(5)' and 'aborted(6)'."

::= { atmLoopbackTestTypes 5 }

atmLoopbackVcLocationId OBJECT-IDENTITY

Expires May 28, 2000

[Page 20]

STATUS current

DESCRIPTION

"This is a loopback test performed on a portion of a designated Vc segment. To perform this test a loopback OAM cell is inserted at a connection point of the designated VC connection (e.g., the end-point or a tandem point) with a value inserted in the Location Identifier ID field of the OAM cell that corresponds to the ATM device where the cell is to be looped back. The loopback cell then travels through the VC connection until it reaches the designated ATM device, where it is looped back to the loopback cell insertion point on the designated VC. Success is achieved if the loopback OAM cell returns to the originating point of insertion within 5 seconds, otherwise, the test fails.

The manager-station performs a loopback test by making use of the testTable defined in [[TestMIB](#)]. In order to run this test the object testType in the testTable shall be set to atmLoopbackVcE2e. The object testTarget points to the row in the atmVclTable in [[RFC2515](#)] corresponding to the VC designated for the test. The object testMoreInfo contains the desired Loopback Location ID.

Before starting a test, a manager-station must first obtain 'ownership' of the entry in the testTable for the interface to be tested (follow procedure defined in [[TestMIB](#)]). Once the manager-station obtains ownership, a loopback test for a given interface can be invoked by first setting up parameters necessary for the loopback test (e.g., set testMoreInfo to AAAABBBBCCCCDDDD and set the testTarget), and then setting the value of testType in the testTable equal to 'atmLoopbackVcSegment.d'. The testRowStatus is used to invoke the atmLoopbackVcLocationId test on the VC with the VPI/VCI corresponding to the testTarget and looped back at loopback location ID= AAAABBBBCCCCDDDD.

After invoking a loopback test, wait for the test completion by polling for the object testResult. A value of 'InProgress(3)' will result if the test is still in progress. Once the test is completed, the object testResult will have a value of 'success(2)' if the loopback OAM cell returned to the originator of the

Expires May 28, 2000

[Page 21]

test within 5 seconds, if not, a value of 'failed(7)' will result. If the ATM system does not support this type of loopback test, then a value of 'notSupported(4)' will be provided. Other possible values for the testResult object are 'unableToRun(5)' and 'aborted(6)'."

::= { atmLoopbackTestTypes 6 }

atmLoopbackVpServiceInternal OBJECT-IDENTITY

STATUS current

DESCRIPTION

"This is a loopback test that the manager requests an agent to perform over the managed resource's internal portion of a designated VP (i.e., between the ingress and egress interfaces of the VP connection). The agent is provided with the Ingress VPI, Egress Interface, and Egress VPI in order to run this internal test. This test may be useful in proxy situations where the proxy agent represents a network. Implementations of this test may be specific to the managed resource. One implementation in a managed network may be as follows, the managed network inserts a segment loopback OAM cell at the network internal segment end-point (corresponding to the ingress connection point) for the designated VP connection. The loopback cell then travels through the network's portion of the VP connection until it reaches the network's connection point to the egress, where it is looped back to the network's cell insertion point on the designated VP. Success is achieved if the loopback OAM cell returns to the originating internal network segment end-point within 5 seconds, otherwise, the test fails.

The manager-station performs a loopback test by making use of the testTable defined in [[TestMIB](#)]. In order to run this test the object testType in the testTable shall be set to atmLoopbackVpServiceInternal, and the object testTarget points to the row in the atmVpCrossConnectTable in [[RFC2515](#)] corresponding to the VP designated for the test.

Before starting a test, a manager-station must first obtain 'ownership' of the entry in the testTable for the interface to be tested (follow procedure defined in [[TestMIB](#)]). Once the manager-station obtains

Expires May 28, 2000

[Page 22]

ownership, a loopback test for a given interface can be invoked by first setting up parameters necessary for the loopback test (e.g., set the testTarget), and then setting the value of testType in the testTable equal to 'atmLoopbackVpServiceInternal'. The testRowStatus is used to invoke the atmLoopbackVpServiceInternal test on the VP crossconnect with the ingress and egress VPI values corresponding to the testTarget.

After invoking a loopback test, wait for the test completion by polling for the object testResult. A value of 'InProgress(3)' will result if the test is still in progress. Once the test is completed, the object testResult will have a value of 'success(2)' if the loopback OAM cell returned to the originator of the test within 5 seconds, if not, a value of 'failed(7)' will result. If the ATM system does not support this type of loopback test, then a value of 'notSupported(4)' will be provided. Other possible values for the testResult object are 'unableToRun(5)' and 'aborted(6)'."

::= { atmLoopbackTestTypes 7 }

atmLoopbackVcServiceInternal OBJECT-IDENTITY

STATUS current

DESCRIPTION

"This is a loopback test that the manager requests an agent to perform over the managed resource's internal portion of a designated VC (i.e., between the ingress and egress interfaces of the VC connection). The agent is provided with the Ingress VPI, Ingress VCI, Egress Interface, Egress VPI, and Egress VCI in order to run this internal test. This test may be useful in proxy situations where the proxy agent represents a network. Implementations of this test may be specific to the managed resource. One implementation in a managed network may be as follows, the managed network inserts a segment loopback OAM cell at the network internal segment end-point (corresponding to the ingress connection point) for the designated VC connection. The loopback cell then travels through the network's portion of the VC connection until it reaches the network's connection point to the egress, where it is looped back to the network's cell insertion point on the designated VC. Success is achieved if the loopback

Expires May 28, 2000

[Page 23]

OAM cell returns to the originating internal network segment end-point within 5 seconds, otherwise, the test fails.

The manager-station performs a loopback test by making use of the testTable defined in [[TestMIB](#)]. In order to run this test the object testType in the testTable shall be set to atmLoopbackVcServiceInternal, and the object testTarget points to the row in the atmVcCrossConnectTable in [[RFC2515](#)] corresponding to the VC designated for the test.

Before starting a test, a manager-station must first obtain 'ownership' of the entry in the testTable for the interface to be tested (follow procedure defined in [[TestMIB](#)]). Once the manager-station obtains ownership, a loopback test for a given interface can be invoked by first setting up parameters necessary for the loopback test (e.g., set the testTarget), and then setting the value of testType in the testTable equal to 'atmLoopbackVcServiceInternal'. The testRowStatus is used to invoke the atmLoopbackVcServiceInternal test on the VC crossconnect with the ingress and egress VPI/VCI values corresponding to the testTarget.

After invoking a loopback test, wait for the test completion by polling for the object testResult. A value of 'InProgress(3)' will result if the test is still in progress. Once the test is completed, the object testResult will have a value of 'success(2)' if the loopback OAM cell returned to the originator of the test within 5 seconds, if not, a value of 'failed(7)' will result. If the ATM system does not support this type of loopback test, then a value of 'notSupported(4)' will be provided. Other possible values for the testResult object are 'unableToRun(5)' and 'aborted(6)'."

::= { atmLoopbackTestTypes 8 }

Expires May 28, 2000

[Page 24]

```
-- *****
-- (2) ATM End-Point Tables

-- This concerns information for interfaces
-- supporting ATM Loopback Tests and includes:
-- 1. ATM VP End-Point Table
-- 2. ATM VC End-Point Table

atmEndptGroup OBJECT IDENTIFIER ::= {
    atmTESTMIBObjects 2}

-- 1. ATM VP End-Point Table

atmVplEndptTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF AtmVplEndptEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "End-point Information for each VP."
    ::= { atmEndptGroup 1 }

atmVplEndptEntry OBJECT-TYPE
    SYNTAX AtmVplEndptEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry with end-point information about the ATM
        VP."
    AUGMENTS { atmVplEntry }
    ::= { atmVplEndptTable 1 }

AtmVplEndptEntry ::=
    SEQUENCE {
        atmVplEndptSegmentEndPoint    INTEGER
    }

atmVplEndptSegmentEndPoint OBJECT-TYPE
    SYNTAX INTEGER {
        isaVplSegmentEndPoint(1),
        notaVplSegmentEndPoint(2)
    }
```

Expires May 28, 2000

[Page 25]

```
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "An indication of whether or not the VP interface
    has been configured to represent a VPC Segment
    End-Point. If the corresponding VP Link is a UNI,
    the value of this object is permanently set to
    isaVplSegmentEndPoint(1). Otherwise, the default
    is set to notaVplSegmentEndPoint(2)."
```

```
::= { atmVplEndptEntry 1 }
```

-- 2. ATM VC End-Point Table

```
atmVclEndptTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF AtmVclEndptEntry
    MAX-ACCESS       not-accessible
    STATUS           current
    DESCRIPTION
        "End-point Information for each VC."
    ::= { atmEndptGroup 2 }
```

```
atmVclEndptEntry OBJECT-TYPE
    SYNTAX  AtmVclEndptEntry
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
        "An entry with end-point information about the ATM
        VC."
    AUGMENTS { atmVclEntry }
    ::= { atmVclEndptTable 1 }
```

```
AtmVclEndptEntry ::=
    SEQUENCE {
        atmVclEndptSegmentEndPoint  INTEGER
    }
```

```
atmVclEndptSegmentEndPoint OBJECT-TYPE
    SYNTAX  INTEGER {
        isaVclSegmentEndPoint(1),
        notaVclSegmentEndPoint(2)
    }
    MAX-ACCESS  read-create
```

Expires May 28, 2000

[Page 26]

```
STATUS    current
DESCRIPTION
    "An indication of whether or not the VC interface
    has been configured to represent a VCC Segment
    End-Point. If the corresponding VC Link is a UNI,
    the value of this object is permanently set to
    isaVclSegmentEndPoint(1). Otherwise, the default
    is set to notaVclSegmentEndPoint(2)."
```

::= { atmVclEndptEntry 1 }

-- *****

-- Conformance Information

```
atmTESTMIBConformance    OBJECT IDENTIFIER ::= {atmTESTMIB 2}

atmTESTMIBGroups          OBJECT IDENTIFIER
                           ::= {atmTESTMIBConformance 1}

atmTESTMIBCompliances     OBJECT IDENTIFIER
                           ::= {atmTESTMIBConformance 2}

-- Compliance Statements

atmTESTMIBCompliance     MODULE-COMPLIANCE
    STATUS                current
    DESCRIPTION
        "The compliance statement for SNMP entities which
        represent ATM interfaces. The compliance statements
        are used to determine if a particular group or object
        applies to hosts, networks/switches, or both."

    MODULE -- this module

        MANDATORY-GROUPS { atmLoopbackGroup }

-- Objects in the ATM Loopback Test Group

OBJECT        atmIfLogicalLoopbackId
MIN-ACCESS    read-only
DESCRIPTION
    "Write access is not required. This object is
    required for ATM systems supporting the
    atmLoopbackVpLocationID and atmLoopbackVcLocationID
    tests."
```

Expires May 28, 2000

[Page 27]

OBJECT atmVplEndptSegmentEndPoint

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required. This object is mandatory for systems that are supporting ATM loopback tests."

OBJECT atmVclEndptSegmentEndPoint

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required. This object is mandatory for systems that are supporting ATM loopback tests."

::= { atmTESTMIBCompliances 1 }

-- *****

-- Units of Conformance

atmLoopbackGroup OBJECT-GROUP

OBJECTS {
 atmIfLogicalLoopbackId,
 atmVplEndptSegmentEndPoint,
 atmVclEndptSegmentEndPoint
}

STATUS current

DESCRIPTION

"A collection of objects providing information for Loopback Tests."

::= { atmTESTMIBGroups 1 }

END

Expires May 28, 2000

[Page 28]

6. Security Considerations

There are a number of management objects defined in this MIB that have a MAX-ACCESS clause of read-write and/or read-create. Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations.

The managed objects in this MIB contain sensitive information since, collectively, they allow the invocation of tests on the managed device.

This MIB contains objects which allow an administrator to perform tests on ATM interfaces. Tests can only be performed when using these objects in conjunction with [\[TestMIB\]](#). Unauthorized access to the associated objects could cause a denial of service, or in combination with other (e.g., physical) security breaches, could cause unauthorized connectivity to a device. The tests defined in this document are not service interrupting.

It is thus important to control even GET access to these objects and possibly to even encrypt the values of these object when sending them over the network via SNMP. Not all versions of SNMP provide features for such a secure environment.

SNMPv1 by itself is not a secure environment. Even if the network itself is secure (for example by using IPSec), even then, there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB.

It is recommended that the implementers consider the security features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model [RFC 2574](#) [[RFC2574](#)] and the View-based Access Control Model [RFC 2575](#) [[RFC2575](#)] is recommended.

It is then a customer/user responsibility to ensure that the SNMP entity giving access to an instance of this MIB, is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

Expires May 28, 2000

[Page 29]

7. Acknowledgments

This document is a product of the AToMMIB Working Group. The authors would like to acknowledge the following persons for their valuable suggestions for this memo

- Dawn Xie
- Umberto Bonollo
- Rod Miller

8. References

- [RFC2571] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", [RFC 2571](#), April 1999
- [RFC1155] Rose, M., and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", STD 16, [RFC 1155](#), May 1990
- [RFC1212] Rose, M., and K. McCloghrie, "Concise MIB Definitions", STD 16, [RFC 1212](#), March 1991
- [RFC1215] M. Rose, "A Convention for Defining Traps for use with the SNMP", [RFC 1215](#), March 1991
- [RFC2578] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M., and S. Waldbusser, "Structure of Management Information Version 2 (SMIv2)", STD 58, [RFC 2578](#), April 1999
- [RFC2579] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M., and S. Waldbusser, "Textual Conventions for SMIv2", STD 58, [RFC 2579](#), April 1999
- [RFC2580] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M., and S. Waldbusser, "Conformance Statements for SMIv2", STD 58, [RFC 2580](#), April 1999
- [RFC1157] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol", STD 15, [RFC 1157](#), May 1990.
- [RFC1901] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Introduction to Community-based SNMPv2", [RFC 1901](#), January 1996.
- [RFC1906] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)", [RFC 1906](#), January 1996.

Expires May 28, 2000

[Page 31]

- [RFC2572] Case, J., Harrington D., Presuhn R., and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", [RFC 2572](#), April 1999
- [RFC2574] Blumenthal, U., and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", [RFC 2574](#), April 1999
- [RFC1905] 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.
- [RFC2573] Levi, D., Meyer, P., and B. Stewart, "SNMPv3 Applications", [RFC 2573](#), April 1999
- [RFC2575] Wijnen, B., Presuhn, R., and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", [RFC 2575](#), April 1999
- [RFC2570] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction to Version 3 of the Internet-standard Network Management Framework", [RFC 2570](#), April 1999
- [RFC2515] Kaj Tesink, "Definitions of Managed Objects for ATM Management", [RFC 2515](#), Bellcore, February 1999.
- [RFC1213] McCloghrie, K., and M. Rose, Editors, "Management Information Base for Network Management of TCP/IP-based internets: MIB-II", STD 17, [RFC 1213](#), Hughes LAN Systems, Performance Systems International, March 1991.
- [RFC2233] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", [RFC2233](#), cisco Systems, FTP Software, November 1997.
- [UNI3.0] ATM Forum, "ATM User-Network Interface, Version 3.0 (UNI 3.0) Specification, Part I", 1994.

Expires May 28, 2000

[Page 32]

- [UNI3.1] ATM Forum, "ATM User-Network Interface, Version 3.1 (UNI 3.1) Specification, Part I", November 1994.
- [I.610] ITU-T Recommendation I.610, "Broadband Integrated Service Digital Network (B-ISDN) Operation and Maintenance Principles and Functions", July 1995.
- [TestMIB] McCloghrie, K., M. Greene, and K. Tesink, "Definitions of Managed Objects for System and Interface Testing", RFCxxxx, cisco Systems, Xedia Corp, Telcordia, ??? 1999.

9. Authors' Addresses

Michael Noto
3Com Corporation
5400 Bayfront Plaza
Santa Clara, CA 95052
Phone: (408) 326-2218
E-mail: mike_noto@3com.com

Kaj Tesink
Telcordia Technologies
331 Newman Springs Road
P.O. Box 7020
Red Bank, NJ 07701-7020
Phone: (732) 758-5254
EMail: kaj@research.telcordia.com

10. RFC Editor and IANA Considerations

Prior to publication of this memo as an RFC, the RFC Editor and IANA are requested replace xxxx below with the RFC number of this document, to make a suitable OBJECT IDENTIFIER assignment for XX below and update the following in the MIB:

"Initial version, published as RFCxxxx"
 ::= { mib-2 XX }

-- ***** NOTE TO THE RFC EDITOR / IANA *****

Expires May 28, 2000

[Page 33]

```
-- * Before this module is put on the standards track
-- * fill out RFCxxxx with the RFC number of this document
-- * assign XX by IANA.
-- * remove this notice from the MIB
```

11. 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.

Expires May 28, 2000

[Page 35]

12. Full Copyright Statement

Copyright (C) The Internet Society (1999). 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.

Table of Contents

1 Status of this Memo	1
2 Abstract	2
3 The SNMP Network Management Framework	2
4 Overview	4
4.1 Background	4
4.2 Terminology	5
4.3 Supported Functions	6
4.3.1 ATM Loopback Testing	6

Expires May 28, 2000

[Page 36]

4.3.2	ATM End-Point Tables	10
5	Definitions	11
6	Security Considerations	29
7	Acknowledgments	30
8	References	31
9	Authors' Addresses	33
10	RFC Editor and IANA Considerations	33
11	Intellectual Property	35
12	Full Copyright Statement	36