Network Working Group
Request for Comments: 3796
Category: Informational

P. Nesser, II
Nesser & Nesser Consulting
A. Bergstrom, Ed.
Ostfold University College

June 2004

Survey of IPv4 Addresses in Currently Deployed IETF Operations & Management Area Standards Track and Experimental Documents

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Abstract

This document seeks to record all usage of IPv4 addresses in currently deployed IETF Operations & Management Area accepted standards. In order to successfully transition from an all IPv4 Internet to an all IPv6 Internet, many interim steps will be taken. One of these steps is the evolution of current protocols that have IPv4 dependencies. It is hoped that these protocols (and their implementations) will be redesigned to be network address independent, but failing that will at least dually support IPv4 and IPv6. To this end, all Standards (Full, Draft, and Proposed), as well as Experimental RFCs, will be surveyed and any dependencies will be documented.

Table of Contents

1. Introduction

This document is part of a set aiming to record all usage of IPv4 addresses in IETF standards. In an effort to have the information in a manageable form, it has been broken into 7 documents conforming to the current IETF areas (Application, Internet, Operations & Management, Routing, Security, Sub-IP and Transport).

For a full introduction, please see the introduction [1].

2. Document Organization

The document is organized as described below:

Sections 3, 4, 5, and 6 each describe the raw analysis of Full, Draft, and Proposed Standards, and Experimental RFCs. Each RFC is discussed in its turn starting with RFC 1 and ending with (around) RFC 3100. The comments for each RFC are "raw" in nature. That is, each RFC is discussed in a vacuum and problems or issues discussed do not "look ahead" to see if the problems have already been fixed.

Section 7 is an analysis of the data presented in Sections 3, 4, 5, and 6. It is here that all of the results are considered as a whole and the problems that have been resolved in later RFCs are correlated.

3. Full Standards

Full Internet Standards (most commonly simply referred to as "Standards") are fully mature protocol specification that are widely implemented and used throughout the Internet.

3.1. RFC 1155 Structure of Management Information

Section 3.2.3.2. IpAddress defines the following:

> This application-wide type represents a 32-bit internet address. It is represented as an OCTET STRING of length 4, in network byte-order.

There are several instances of the use of this definition in the rest of the document.

3.2. RFC 1212 Concise MIB definitions

In section 4.1.6 IpAddress is defined as:

> (6) IpAddress-valued: 4 sub-identifiers, in the familiar a.b.c.d notation.

3.3. RFC 1213 Management Information Base

There are far too many instances of IPv4 addresses is this document to enumerate here. The particular object groups that are affected are the IP group, the ICMP group, the TCP group, the UDP group, and the EGP group.

3.4. RFC 2578 Structure of Management Information Version 2 (SMIv2)

Section 7.1.5 defines the IpAddress data type:

> The IpAddress type represents a 32-bit internet address. It is represented as an OCTET STRING of length 4, in network byte-order.
>
> Note that the IpAddress type is a tagged type for historical reasons. Network addresses should be represented using an invocation of the TEXTUAL-CONVENTION macro.

Note the deprecated status of this type; see RFC 3291 for details on the replacement TEXTUAL-CONVENTION definitions.

3.5. RFC 2579 Textual Conventions for SMIv2

There are no IPv4 dependencies in this specification.

3.6. RFC 2580 Conformance Statements for SMIv2

There are no IPv4 dependencies in this specification.

3.7. RFC 2819 Remote Network Monitoring Management Information Base

There are no IPv4 dependencies in this specification.

3.8. RFC 3411 An Architecture for Describing SNMP Management Frameworks

There are no IPv4 dependencies in this specification.

3.9. RFC 3412 Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)

There are no IPv4 dependencies in this specification.

3.10. RFC 3413 SNMP Applications

There are no IPv4 dependencies in this specification.

3.11. RFC 3414 User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)

There are no IPv4 dependencies in this specification.

3.12. RFC 3415 View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)

There are no IPv4 dependencies in this specification.

3.13. RFC 3416 Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMP)

Section 4.2.2.1., Example of Table Traversal, and Section 4.2.3.1., Another Example of Table Traversal, both use objects from MIB2 whose data contains IPv4 addresses. Other than their use in these example sections, there are no IPv4 dependencies in this specification.

3.14. RFC 3417 Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMP)

Section 2 Definitions contains the following definition:

```
SnmpUDPAddress ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "1d.1d.1d.1d/2d"
    STATUS       current
    DESCRIPTION
            "Represents a UDP address:

                octets   contents        encoding
                1-4     IP-address      network-byte order
                5-6     UDP-port        network-byte order
            "
   SYNTAX       OCTET STRING (SIZE (6))
```

Section 8.1, Usage Example, also contains examples which uses IPv4 address, but it has no significance in the operation of the specification.

3.15. RFC 3418 Management Information Base for Version 2 of the Simple Network Management Protocol (SNMP)

There are no IPv4 dependencies in this specification.

4. Draft Standards

Draft Standards represent the penultimate standard level in the IETF. A protocol can only achieve draft standard when there are multiple, independent, interoperable implementations. Draft Standards are usually quite mature and widely used.

4.1. RFC 1493 Definitions of Managed Objects for Bridges

There are no IPv4 dependencies in this specification.

4.2. RFC 1559 DECnet Phase IV MIB Extensions

There are no IPv4 dependencies in this specification.

4.3. RFC 1657 Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP-4) using SMIv2

The MIB defined in this RFC deals with objects in a BGP4 based routing system and therefore contain many objects that are limited by the IpAddress 32-bit value defined in MIB2. Clearly the values of this MIB are limited to IPv4 addresses. No update is needed, although a new MIB should be defined for BGP4+ to allow management of IPv6 addresses and routes.

4.4. RFC 1658 Definitions of Managed Objects for Character Stream Devices using SMIv2

There are no IPv4 dependencies in this specification.

4.5. RFC 1659 Definitions of Managed Objects for RS-232-like Hardware Devices using SMIv2

There are no IPv4 dependencies in this specification.

4.6. RFC 1660 Definitions of Managed Objects for Parallel-printer-like Hardware Devices using SMIv2

There are no IPv4 dependencies in this specification.

4.7. RFC 1694 Definitions of Managed Objects for SMDS Interfaces using SMIv2

This MIB module definition defines the following subtree:

```
ipOverSMDS OBJECT IDENTIFIER ::= { smdsApplications 1 }

-- Although the objects in this group are read-only, at the
-- agent's discretion they may be made read-write so that the
-- management station, when appropriately authorized, may
-- change the addressing information related to the
-- configuration of a logical IP subnetwork implemented on
-- top of SMDS.

-- This table is necessary to support RFC1209 (IP-over-SMDS)
-- and gives information on the Group Addresses and ARP
-- Addresses used in the Logical IP subnetwork.
-- One SMDS address may be associated with multiple IP
-- addresses.  One SNI may be associated with multiple LISs.

ipOverSMDSTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF IpOverSMDSEntry
    MAX-ACCESS  not-accessible
```

```
        STATUS      current
        DESCRIPTION
           "The table of addressing information relevant to
           this entity's IP addresses."
        ::= { ipOverSMDS 1 }

    ipOverSMDSEntry OBJECT-TYPE
        SYNTAX      IpOverSMDSEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
           "The addressing information for one of this
           entity's IP addresses."
        INDEX   { ipOverSMDSIndex, ipOverSMDSAddress }
        ::= { ipOverSMDSTable 1 }

    IpOverSMDSEntry ::=
        SEQUENCE {
           ipOverSMDSIndex       IfIndex,
           ipOverSMDSAddress     IpAddress,
           ipOverSMDSHA          SMDSAddress,
           ipOverSMDSLISGA       SMDSAddress,
           ipOverSMDSARPReq      SMDSAddress
           }

    ipOverSMDSIndex OBJECT-TYPE
        SYNTAX      IfIndex
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
           "The value of this object identifies the
           interface for which this entry contains management
           information. "
        ::= { ipOverSMDSEntry 1 }

    ipOverSMDSAddress OBJECT-TYPE
         SYNTAX      IpAddress
         MAX-ACCESS  read-only
         STATUS      current
         DESCRIPTION
           "The IP address to which this entry's addressing
           information pertains."
        ::= { ipOverSMDSEntry 2 }

    ipOverSMDSHA OBJECT-TYPE
        SYNTAX      SMDSAddress
        MAX-ACCESS  read-only
        STATUS      current
```

```
        DESCRIPTION
           "The SMDS Individual address of the IP station."
        ::= { ipOverSMDSEntry 3 }

    ipOverSMDSLISGA OBJECT-TYPE
        SYNTAX      SMDSAddress
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
           "The SMDS Group Address that has been configured
           to identify the SMDS Subscriber-Network Interfaces
           (SNIs) of all members of the Logical IP Subnetwork
           (LIS) connected to the network supporting SMDS."
        ::= { ipOverSMDSEntry 4 }

    ipOverSMDSARPReq OBJECT-TYPE
        SYNTAX      SMDSAddress
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
           "The SMDS address (individual or group) to which
           ARP Requests are to be sent."
        ::= { ipOverSMDSEntry 5 }
```

Although these object definitions are intended for IPv4 addresses, a similar MIB can be defined for IPv6 addressing.

4.8. RFC 1724 RIP Version 2 MIB Extension

As expected, this RFC is filled with IPv4 dependencies since it defines a MIB module for an IPv4-only routing protocol. A new MIB for RIPng is required.

4.9. RFC 1748 IEEE 802.5 MIB using SMIv2

There are no IPv4 dependencies in this specification.

4.10. RFC 1850 OSPF Version 2 Management Information Base

This MIB defines managed objects for OSPFv2 which is a protocol used to exchange IPv4 routing information. Since OSPFv2 is limited to IPv4 addresses, a new MIB is required to support a new version of OSPF that is IPv6 aware.

4.11. RFC 2115 Management Information Base for Frame Relay DTEs Using SMIv2

This specification has several examples of how IPv4 addresses might be mapped to Frame Relay DLCIs. Other than those examples there are no IPv4 dependencies in this specification.

4.12. RFC 2790 Host Resources MIB

There are no IPv4 dependencies in this specification.

4.13. RFC 2863 The Interfaces Group MIB

There are no IPv4 dependencies in this specification. There is some discussion in one object definition about an interface performing a self test, but the object itself is IP version independent.

4.14. RFC 3592 Definitions of Managed Objects for the Synchronous Optical Network/Synchronous Digital Hierarchy (SONET/SDH)

There are no IPv4 dependencies in this specification.

4.15. RFC 3593 Textual Conventions for MIB Modules Using Performance History Based on 15 Minute Intervals

There are no IPv4 dependencies in this specification.

5. Proposed Standards

Proposed Standards are introductory level documents. There are no requirements for even a single implementation. In many cases, Proposed are never implemented or advanced in the IETF standards process. They therefore are often just proposed ideas that are presented to the Internet community. Sometimes flaws are exposed or they are one of many competing solutions to problems. In these later cases, no discussion is presented as it would not serve the purpose of this discussion.

5.1. RFC 1239 Reassignment of experimental MIBs to standard MIBs

There are no IPv4 dependencies in this specification.

5.2. RFC 1269 Definitions of Managed Objects for the Border Gateway Protocol: Version 3

The use of BGP3 has been deprecated and is not discussed.

5.3. RFC 1285 FDDI Management Information Base

There are no IPv4 dependencies in this specification.

5.4. RFC 1381 SNMP MIB Extension for X.25 LAPB

There are no IPv4 dependencies in this specification.

5.5. RFC 1382 SNMP MIB Extension for the X.25 Packet Layer

There are no IPv4 dependencies in this specification.

5.6. RFC 1414 Identification MIB

There are no IPv4 dependencies in this specification.

5.7. RFC 1418 SNMP over OSI

There are no IPv4 dependencies in this specification.

5.8. RFC 1419 SNMP over AppleTalk

There are no IPv4 dependencies in this specification.

5.9. RFC 1420 SNMP over IPX

There are no IPv4 dependencies in this specification.

5.10. RFC 1461 SNMP MIB extension for Multiprotocol Interconnect over X.25

The following objects are defined in Section 4, Definitions:

```
mioxPleLastFailedEnAddr OBJECT-TYPE
        SYNTAX  OCTET STRING (SIZE(2..128))
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The last Encapsulated address that failed
                to find a corresponding X.121 address and
                caused mioxPleEnAddrToX121LkupFlrs to be
                incremented.  The first octet of this object
                contains the encapsulation type, the
                remaining octets contain the address of that
                type that failed.  Thus for an IP address,
                the length will be five octets, the first
                octet will contain 204 (hex CC), and the
                last four octets will contain the IP
```

```
                address.  For a snap encapsulation, the
                first byte would be 128 (hex 80) and the
                rest of the octet string would have the snap
                header."
        ::= { mioxPleEntry 4 }

   mioxPeerEnAddr  OBJECT-TYPE
        SYNTAX    OCTET STRING (SIZE (0..128))
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The Encapsulation address of the remote
                host mapped by this table entry.  A length
                of zero indicates the remote IP address is
                unknown or unspecified for use as a PLE
                default.

                The first octet of this object contains the
                encapsulation type, the remaining octets
                contain an address of that type.  Thus for
                an IP address, the length will be five
                octets, the first octet will contain 204
                (hex CC), and the last four octets will
                contain the IP address.  For a snap
                encapsulation, the first byte would be 128
                (hex 80) and the rest of the octet string
                would have the snap header."
        DEFVAL { ''h }
        ::= { mioxPeerEntry 7 }

mioxPeerEncType OBJECT-TYPE
        SYNTAX  INTEGER (0..256)
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The value of the encapsulation type.  For
                IP encapsulation this will have a value of
                204 (hex CC).  For SNAP encapsulated
                packets, this will have a value of 128 (hex
                80).  For CLNP, ISO 8473, this will have a
                value of 129 (hex 81).  For ES-ES, ISO 9542,
                this will have a value of 130 (hex 82).  A
                value of 197 (hex C5) identifies the Blacker
                X.25 encapsulation.  A value of 0,
                identifies the Null encapsulation.

                This value can only be written when the
                mioxPeerStatus object with the same
```

```
                    mioxPeerIndex has a value of underCreation.
                    Setting this object to a value of 256
                    deletes the entry.  When deleting an entry,
                    all other entries in the mioxPeerEncTable
                    with the same mioxPeerIndex and with an
                    mioxPeerEncIndex higher then the deleted
                    entry, will all have their mioxPeerEncIndex
                    values decremented by one."
            ::= { mioxPeerEncEntry 2 }
```

Updated values of the first byte of these objects can be defined to support IPv6 addresses.

5.11. RFC 1471 The Definitions of Managed Objects for the Link Control Protocol of the Point-to-Point Protocol

There are no IPv4 dependencies in this specification.

5.12. RFC 1472 The Definitions of Managed Objects for the Security Protocols of the Point-to-Point Protocol

There are no IPv4 dependencies in this specification.

5.13. RFC 1473 The Definitions of Managed Objects for the IP Network Control Protocol of the Point-to-Point Protocol

This MIB module is targeted specifically at IPv4 over PPP. A new MIB module would need to be defined to support IPv6 over PPP.

5.14. RFC 1474 The Definitions of Managed Objects for the Bridge Network Control Protocol of the Point-to-Point Protocol

There are no IPv4 dependencies in this specification.

5.15. RFC 1512 FDDI Management Information Base

There are no IPv4 dependencies in this specification.

5.16. RFC 1513 Token Ring Extensions to the Remote Network Monitoring MIB

There are no IPv4 dependencies in this specification.

5.17. RFC 1525 Definitions of Managed Objects for Source Routing Bridges

There are no IPv4 dependencies in this specification.

5.18. RFC 1628 UPS Management Information Base

There are no IPv4 dependencies in this specification.

5.19. RFC 1666 Definitions of Managed Objects for SNA NAUs using SMIv2

There are no IPv4 dependencies in this specification.

5.20. RFC 1696 Modem Management Information Base (MIB) using SMIv2

There are no IPv4 dependencies in this specification.

5.21. RFC 1697 Relational Database Management System (RDBMS) Management Information Base (MIB) using SMIv2

There are no IPv4 dependencies in this specification.

5.22. RFC 1742 AppleTalk Management Information Base II

The following objects are defined:

```
KipEntry ::= SEQUENCE {
     kipNetStart     ATNetworkNumber,
     kipNetEnd       ATNetworkNumber,
     kipNextHop      IpAddress,
     kipHopCount     INTEGER,
     kipBCastAddr    IpAddress,
     kipCore         INTEGER,
     kipType         INTEGER,
     kipState        INTEGER,
     kipShare        INTEGER,
     kipFrom         IpAddress
 }

 kipNextHop OBJECT-TYPE
     SYNTAX IpAddress
     ACCESS read-write
     STATUS mandatory
     DESCRIPTION
         "The IP address of the next hop in the route to this
         entry's destination network."
     ::= { kipEntry 3 }

 kipBCastAddr OBJECT-TYPE
     SYNTAX IpAddress
     ACCESS read-write
     STATUS mandatory
     DESCRIPTION
```

```
            "The form of the IP address used to broadcast on this
            network."
        ::= { kipEntry 5 }

    kipFrom OBJECT-TYPE
        SYNTAX IpAddress
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            "The IP address from which the routing entry was
            learned via the AA protocol.  If this entry was not
            created via the AA protocol, it should contain IP
            address 0.0.0.0."
        ::= { kipEntry 10 }
```

5.23. RFC 1747 Definitions of Managed Objects for SNA Data Link Control (SDLC) using SMIv2

There are no IPv4 dependencies in this specification.

5.24. RFC 1749 IEEE 802.5 Station Source Routing MIB using SMIv2

There are no IPv4 dependencies in this specification.

5.25. RFC 1759 Printer MIB

There are no IPv4 dependencies in this specification.

5.26. RFC 2006 The Definitions of Managed Objects for IP Mobility Support using SMIv2

This document defines a MIB for the Mobile IPv4. Without enumeration, let it be stated that a new MIB for IPv6 Mobility is required.

5.27. RFC 2011 SNMPv2 Management Information Base for the Internet Protocol using SMIv2

Approximately 1/3 of the objects defined in this document are IPv4-dependent. New objects need to be defined to support IPv6.

5.28. RFC 2012 SNMPv2 Management Information Base for the Transmission Control Protocol using SMIv2

A number of object definitions in this MIB assumes IPv4 addresses, as is noted in the note reproduced below:

IESG Note:

> The IP, UDP, and TCP MIB modules currently support only IPv4. These three modules use the IpAddress type defined as an OCTET STRING of length 4 to represent the IPv4 32-bit internet addresses. (See RFC 1902, SMI for SNMPv2.) They do not support the new 128-bit IPv6 internet addresses.

5.29. RFC 2013 SNMPv2 Management Information Base for the User Datagram Protocol using SMIv2

A number of object definitions in this MIB assumes IPv4 addresses, as is noted in the note reproduced below:

IESG Note:

> The IP, UDP, and TCP MIB modules currently support only IPv4. These three modules use the IpAddress type defined as an OCTET STRING of length 4 to represent the IPv4 32-bit internet addresses. (See RFC 1902, SMI for SNMPv2.) They do not support the new 128-bit IPv6 internet addresses.

5.30. RFC 2020 IEEE 802.12 Interface MIB

There are no IPv4 dependencies in this specification.

5.31. RFC 2021 Remote Network Monitoring Management Information Base Version 2 using SMIv2

The following objects are defined:

```
addressMapNetworkAddress OBJECT-TYPE
    SYNTAX      OCTET STRING
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The network address for this relation.

        This is represented as an octet string with
        specific semantics and length as identified
        by the protocolDirLocalIndex component of the
        index.
```

```
            For example, if the protocolDirLocalIndex indicates an
            encapsulation of ip, this object is encoded as a length
            octet of 4, followed by the 4 octets of the ip address,
            in network byte order."
        ::= { addressMapEntry 2 }

    nlHostAddress OBJECT-TYPE
        SYNTAX      OCTET STRING
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The network address for this nlHostEntry.

            This is represented as an octet string with
            specific semantics and length as identified
            by the protocolDirLocalIndex component of the index.

            For example, if the protocolDirLocalIndex indicates an
            encapsulation of ip, this object is encoded as a length
            octet of 4, followed by the 4 octets of the ip address,
            in network byte order."
        ::= { nlHostEntry 2 }

    nlMatrixSDSourceAddress OBJECT-TYPE
        SYNTAX      OCTET STRING
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The network source address for this nlMatrixSDEntry.

            This is represented as an octet string with
            specific semantics and length as identified
            by the protocolDirLocalIndex component of the index.

            For example, if the protocolDirLocalIndex indicates an
            encapsulation of ip, this object is encoded as a length
            octet of 4, followed by the 4 octets of the ip address,
            in network byte order."
        ::= { nlMatrixSDEntry 2 }

    nlMatrixSDDestAddress OBJECT-TYPE
        SYNTAX      OCTET STRING
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The network destination address for this
            nlMatrixSDEntry.
```

```
            This is represented as an octet string with
            specific semantics and length as identified
            by the protocolDirLocalIndex component of the index.

            For example, if the protocolDirLocalIndex indicates an
            encapsulation of ip, this object is encoded as a length
            octet of 4, followed by the 4 octets of the ip address,
            in network byte order."
        ::= { nlMatrixSDEntry 3 }

    nlMatrixDSSourceAddress OBJECT-TYPE
        SYNTAX      OCTET STRING
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The network source address for this nlMatrixDSEntry.

            This is represented as an octet string with
            specific semantics and length as identified
            by the protocolDirLocalIndex component of the index.

            For example, if the protocolDirLocalIndex indicates an
            encapsulation of ip, this object is encoded as a length
            octet of 4, followed by the 4 octets of the ip address,
            in network byte order."
        ::= { nlMatrixDSEntry 2 }

    nlMatrixDSDestAddress OBJECT-TYPE
        SYNTAX      OCTET STRING
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The network destination address for this
            nlMatrixDSEntry.

            This is represented as an octet string with
            specific semantics and length as identified
            by the protocolDirLocalIndex component of the index.

            For example, if the protocolDirLocalIndex indicates an
            encapsulation of ip, this object is encoded as a length
            octet of 4, followed by the 4 octets of the ip address,
            in network byte order."
        ::= { nlMatrixDSEntry 3 }

    nlMatrixTopNSourceAddress OBJECT-TYPE
        SYNTAX     OCTET STRING
        MAX-ACCESS read-only
```

```
        STATUS     current
        DESCRIPTION
            "The network layer address of the source host in this
            conversation.

            This is represented as an octet string with
            specific semantics and length as identified
            by the associated nlMatrixTopNProtocolDirLocalIndex.

            For example, if the protocolDirLocalIndex indicates an
            encapsulation of ip, this object is encoded as a length
            octet of 4, followed by the 4 octets of the ip address,
            in network byte order."
        ::= { nlMatrixTopNEntry 3 }

    nlMatrixTopNDestAddress OBJECT-TYPE
        SYNTAX     OCTET STRING
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The network layer address of the destination host in this
            conversation.

            This is represented as an octet string with
            specific semantics and length as identified
            by the associated nlMatrixTopNProtocolDirLocalIndex.

            For example, if the nlMatrixTopNProtocolDirLocalIndex
            indicates an encapsulation of ip, this object is encoded as a
            length octet of 4, followed by the 4 octets of the ip
            address, in network byte order."
        ::= { nlMatrixTopNEntry 4 }

    alMatrixTopNSourceAddress OBJECT-TYPE
        SYNTAX     OCTET STRING
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The network layer address of the source host in this
            conversation.
            This is represented as an octet string with
            specific semantics and length as identified
            by the associated alMatrixTopNProtocolDirLocalIndex.

            For example, if the alMatrixTopNProtocolDirLocalIndex
            indicates an encapsulation of ip, this object is encoded as a
            length octet of 4, followed by the 4 octets of the
            ip address, in network byte order."
```

```
        ::= { alMatrixTopNEntry 3 }

    alMatrixTopNDestAddress OBJECT-TYPE
        SYNTAX     OCTET STRING
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The network layer address of the destination host in this
            conversation.

            This is represented as an octet string with
            specific semantics and length as identified
            by the associated alMatrixTopNProtocolDirLocalIndex.

            For example, if the alMatrixTopNProtocolDirLocalIndex
            indicates an encapsulation of ip, this object is encoded as a
            length octet of 4, followed by the 4 octets of the ip
            address, in network byte order."
        ::= { alMatrixTopNEntry 4 }

    trapDestProtocol OBJECT-TYPE
        SYNTAX     INTEGER {
                        ip(1),
                        ipx(2)
                    }
        MAX-ACCESS read-create
        STATUS     current
        DESCRIPTION
            "The protocol with which to send this trap."
        ::= { trapDestEntry 3 }

    trapDestAddress  OBJECT-TYPE
        SYNTAX     OCTET STRING
        MAX-ACCESS read-create
        STATUS     current
        DESCRIPTION
            "The address to send traps on behalf of this entry.

            If the associated trapDestProtocol object is equal to ip(1),
            the encoding of this object is the same as the snmpUDPAddress
            textual convention in [RFC1906]:
              -- for a SnmpUDPAddress of length 6:
              --
              -- octets   contents        encoding
              --  1-4     IP-address      network-byte order
              --  5-6     UDP-port        network-byte order

            If the associated trapDestProtocol object is equal to ipx(2),
```

```
            the encoding of this object is the same as the snmpIPXAddress
            textual convention in [RFC1906]:
              -- for a SnmpIPXAddress of length 12:
              --
              -- octets   contents            encoding
              --  1-4     network-number      network-byte order
              --  5-10    physical-address    network-byte order
              -- 11-12    socket-number       network-byte order

            This object may not be modified if the associated
            trapDestStatus object is equal to active(1)."
        ::= { trapDestEntry 4 }
```

All of the object definitions above (except trapDestProtocol) mention only IPv4 addresses. However, since they use a SYNTAX of OCTET STRING, they should work fine for IPv6 addresses. A new legitimate value of trapDestProtocol (i.e., SYNTAX addition of ipv6(3) should make this specification functional for IPv6.

5.32. RFC 2024 Definitions of Managed Objects for Data Link Switching using SMIv2

The following textual conventions are defined:

```
TAddress ::= TEXTUAL-CONVENTION
    STATUS  current
    DESCRIPTION
       "Denotes a transport service address.
        For dlswTCPDomain, a TAddress is 4 octets long,
        containing the IP-address in network-byte order."
    SYNTAX  OCTET STRING (SIZE (0..255))

-- DLSw over TCP
dlswTCPDomain  OBJECT IDENTIFIER ::= { dlswDomains 1 }
-- for an IP address of length 4:
--
-- octets   contents        encoding
--  1-4     IP-address      network-byte order
--
DlswTCPAddress ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "1d.1d.1d.1d"
    STATUS       current
    DESCRIPTION
            "Represents the IP address of a DLSw which uses
             TCP as a transport protocol."
    SYNTAX       OCTET STRING (SIZE (4))
```

Additionally there are many object definitions that use a SYNTAX of TAddress within the document. Interestingly the SYNTAX for TAddress is an OCTET string of up to 256 characters. It could easily accommodate a similar hybrid format for IPv6 addresses.

A new OID to enhance functionality for DlswTCPAddress could be added to support IPv6 addresses.

5.33. RFC 2051 Definitions of Managed Objects for APPC using SMIv2

There are no IPv4 dependencies in this specification.

5.34. RFC 2096 IP Forwarding Table MIB

The MIB module's main conceptual table ipCidrRouteTable uses IPv4 addresses as index objects and is therefore incapable of representing an IPv6 forwarding information base. A new conceptual table needs to be defined to support IPv6 addresses.

5.35. RFC 2108 Definitions of Managed Objects for IEEE 802.3 Repeater Devices using SMIv2 802

There are no IPv4 dependencies in this specification.

5.36. RFC 2127 ISDN Management Information Base using SMIv2

There are no IPv4 dependencies in this specification.

5.37. RFC 2128 Dial Control Management Information Base using SMIv2

There are no IPv4 dependencies in this specification.

5.38. RFC 2206 RSVP Management Information Base using SMIv2

All of the relevant object definitions in this MIB have options for both IPv4 and IPv6. There are no IPv4 dependencies in this specification.

5.39. RFC 2213 Integrated Services Management Information Base using SMIv2

This MIB is IPv6 aware and therefore there are no IPv4 dependencies in this specification.

5.40. RFC 2214 Integrated Services Management Information Base Guaranteed Service Extensions using SMIv2

There are no IPv4 dependencies in this specification.

5.41. RFC 2232 Definitions of Managed Objects for DLUR using SMIv2

There are no IPv4 dependencies in this specification.

5.42. RFC 2238 Definitions of Managed Objects for HPR using SMIv2

There are no IPv4 dependencies in this specification.

5.43. RFC 2266 Definitions of Managed Objects for IEEE 802.12 Repeater Devices

There are no IPv4 dependencies in this specification.

5.44. RFC 2287 Definitions of System-Level Managed Objects for Applications

There are no IPv4 dependencies in this specification.

5.45. RFC 2320 Definitions of Managed Objects for Classical IP and ARP Over ATM Using SMIv2 (IPOA-MIB)

This MIB is wholly dependent on IPv4. A new MIB for IPv6 is required to provide the same functionality.

5.46. RFC 2417 Definitions of Managed Objects for Multicast over UNI 3.0/3.1 based ATM Networks

This MIB is wholly dependent on IPv4. A new MIB for IPv6 is required to provide the same functionality.

5.47. RFC 2452 IP Version 6 Management Information Base for the Transmission Control Protocol

This RFC documents a soon to be obsoleted IPv6 MIB and is not considered in this discussion.

5.48. RFC 2454 IP Version 6 Management Information Base for the User Datagram Protocol

This RFC documents a soon to be obsoleted IPv6 MIB and is not considered in this discussion.

5.49. RFC 2455 Definitions of Managed Objects for APPN

There are no IPv4 dependencies in this specification.

5.50. RFC 2456 Definitions of Managed Objects for APPN TRAPS

There are no IPv4 dependencies in this specification.

5.51. RFC 2457 Definitions of Managed Objects for Extended Border Node

There are no IPv4 dependencies in this specification.

5.52. RFC 2465 Management Information Base for IP Version 6: Textual Conventions and General Group

This RFC documents a soon to be obsoleted IPv6 MIB and is not considered in this discussion.

5.53. RFC 2466 Management Information Base for IP Version 6: ICMPv6 Group

This RFC documents a soon to be obsoleted IPv6 MIB and is not considered in this discussion.

5.54. RFC 2494 Definitions of Managed Objects for the DS0 and DS0 Bundle Interface Type

There are no IPv4 dependencies in this specification.

5.55. RFC 2495 Definitions of Managed Objects for the DS1, E1, DS2 and E2 Interface Types

There are no IPv4 dependencies in this specification.

5.56. RFC 2496 Definitions of Managed Object for the DS3/E3 Interface Type

There are no IPv4 dependencies in this specification.

5.57. RFC 2512 Accounting Information for ATM Networks

There are no IPv4 dependencies in this specification.

5.58. RFC 2513 Managed Objects for Controlling the Collection and Storage of Accounting Information for Connection-Oriented Networks

There are no IPv4 dependencies in this specification.

5.59. RFC 2514 Definitions of Textual Conventions and OBJECT-IDENTITIES for ATM Management

There are no IPv4 dependencies in this specification.

5.60. RFC 2515 Definitions of Managed Objects for ATM Management

This MIB defines the following objects:

```
AtmInterfaceConfEntry    ::= SEQUENCE  {
     atmInterfaceMaxVpcs             INTEGER,
     atmInterfaceMaxVccs             INTEGER,
     atmInterfaceConfVpcs            INTEGER,
     atmInterfaceConfVccs            INTEGER,
     atmInterfaceMaxActiveVpiBits    INTEGER,
     atmInterfaceMaxActiveVciBits    INTEGER,
     atmInterfaceIlmiVpi             AtmVpIdentifier,
     atmInterfaceIlmiVci             AtmVcIdentifier,
     atmInterfaceAddressType         INTEGER,
     atmInterfaceAdminAddress        AtmAddr,
     atmInterfaceMyNeighborIpAddress IpAddress,
     atmInterfaceMyNeighborIfName    DisplayString,
     atmInterfaceCurrentMaxVpiBits   INTEGER,
     atmInterfaceCurrentMaxVciBits   INTEGER,
     atmInterfaceSubscrAddress       AtmAddr
          }

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

```
   atmInterfaceConfGroup2    OBJECT-GROUP
          OBJECTS {
                atmInterfaceMaxVpcs, atmInterfaceMaxVccs,
                atmInterfaceConfVpcs, atmInterfaceConfVccs,
                atmInterfaceMaxActiveVpiBits,
                atmInterfaceMaxActiveVciBits,
                atmInterfaceIlmiVpi,
                atmInterfaceIlmiVci,
                atmInterfaceMyNeighborIpAddress,
                atmInterfaceMyNeighborIfName,
                atmInterfaceCurrentMaxVpiBits,
                atmInterfaceCurrentMaxVciBits,
                atmInterfaceSubscrAddress }
          STATUS     current
          DESCRIPTION
            "A collection of objects providing configuration
             information about an ATM interface."
          ::= { atmMIBGroups 10 }
```

Clearly a subsequent revision of this MIB module should define equivalent IPv6 objects.

5.61. RFC 2561 Base Definitions of Managed Objects for TN3270E Using SMIv2

The document states:

The MIB defined by this memo supports use of both IPv4 and IPv6 addressing.

This specification is both IPv4 and IPv6 aware.

5.62. RFC 2562 Definitions of Protocol and Managed Objects for TN3270E Response Time Collection Using SMIv2

This MIB module inherits IP version-independence by virtue of importing the appropriate definitions from RFC 2561.

5.63. RFC 2564 Application Management MIB

The following textual convention is defined:

```
   ApplTAddress ::= TEXTUAL-CONVENTION
       STATUS       current
       DESCRIPTION
             "Denotes a transport service address.

             For snmpUDPDomain, an ApplTAddress is 6 octets long,
```

```
            the initial 4 octets containing the IP-address in
            network-byte order and the last 2 containing the UDP
            port in network-byte order.  Consult 'Transport Mappings
            for Version 2 of the Simple Network Management Protocol
            (SNMPv2)' for further information on snmpUDPDomain."
        SYNTAX       OCTET STRING (SIZE (0..255))
```

A new TC should be defined to handle IPv6 addresses.

5.64. RFC 2584 Definitions of Managed Objects for APPN/HPR in IP Networks

Many of the object definitions described in this document assume the use of the IPv4 only TOS header bits. It is therefore IPv4-only in nature and will not support IPv6.

5.65. RFC 2594 Definitions of Managed Objects for WWW Services

There are no IPv4 dependencies in this specification.

5.66. RFC 2605 Directory Server Monitoring MIB

There are no IPv4 dependencies in this specification.

5.67. RFC 2613 Remote Network Monitoring MIB Extensions for Switched Networks Version 1.0

There are no IPv4 dependencies in this specification.

5.68. RFC 2618 RADIUS Authentication Client MIB

This RFC defines the following objects:

```
RadiusAuthServerEntry ::= SEQUENCE {
      radiusAuthServerIndex                           Integer32,
      radiusAuthServerAddress                         IpAddress,
      radiusAuthClientServerPortNumber                Integer32,
      radiusAuthClientRoundTripTime                   TimeTicks,
      radiusAuthClientAccessRequests                  Counter32,
      radiusAuthClientAccessRetransmissions           Counter32,
      radiusAuthClientAccessAccepts                   Counter32,
      radiusAuthClientAccessRejects                   Counter32,
      radiusAuthClientAccessChallenges                Counter32,
      radiusAuthClientMalformedAccessResponses        Counter32,
      radiusAuthClientBadAuthenticators               Counter32,
      radiusAuthClientPendingRequests                   Gauge32,
      radiusAuthClientTimeouts                        Counter32,
      radiusAuthClientUnknownTypes                    Counter32,
```

```
      radiusAuthClientPacketsDropped                    Counter32
   }

   radiusAuthServerAddress OBJECT-TYPE
         SYNTAX     IpAddress
         MAX-ACCESS read-only
         STATUS     current
         DESCRIPTION
               "The IP address of the RADIUS authentication server
                referred to in this table entry."
         ::= { radiusAuthServerEntry 2 }
```

There needs to be an update to allow an IPv6 based object for this value.

5.69. RFC 2619 RADIUS Authentication Server MIB

This MIB defines the followings objects:

```
   RadiusAuthClientEntry ::= SEQUENCE {
          radiusAuthClientIndex                          Integer32,
          radiusAuthClientAddress                        IpAddress,
          radiusAuthClientID                       SnmpAdminString,
          radiusAuthServAccessRequests                   Counter32,
          radiusAuthServDupAccessRequests                Counter32,
          radiusAuthServAccessAccepts                    Counter32,
          radiusAuthServAccessRejects                    Counter32,
          radiusAuthServAccessChallenges                 Counter32,
          radiusAuthServMalformedAccessRequests          Counter32,
          radiusAuthServBadAuthenticators                Counter32,
          radiusAuthServPacketsDropped                   Counter32,
          radiusAuthServUnknownTypes                     Counter32
   }

   radiusAuthClientAddress OBJECT-TYPE
          SYNTAX     IpAddress
          MAX-ACCESS read-only
          STATUS     current
          DESCRIPTION
                "The NAS-IP-Address of the RADIUS authentication client
                 referred to in this table entry."
          ::= { radiusAuthClientEntry 2 }
```

This object needs to be deprecated and replaced by one that supports both IPv4 and IPv6 addresses.

5.70. RFC 2622 Routing Policy Specification Language (RPSL)

The only objects in the version of RPSL that deal with IP addresses are defined as:

<ipv4-address> An IPv4 address is represented as a sequence of four integers in the range from 0 to 255 separated by the character dot ".". For example, 128.9.128.5 represents a valid IPv4 address. In the rest of this document, we may refer to IPv4 addresses as IP addresses.

<address-prefix> An address prefix is represented as an IPv4 address followed by the character slash "/" followed by an integer in the range from 0 to 32. The following are valid address prefixes: 128.9.128.5/32, 128.9.0.0/16, 0.0.0.0/0; and the following address prefixes are invalid: 0/0, 128.9/16 since 0 or 128.9 are not strings containing four integers.

There seems to be an awareness of IPv6 because of the terminology but it is not specifically defined. Therefore additional objects for IPv6 addresses and prefixes need to be defined.

5.71. RFC 2662 Definitions of Managed Objects for the ADSL Lines

There are no IPv4 dependencies in this specification.

5.72. RFC 2667 IP Tunnel MIB

The Abstract of this document says:

> This memo defines a Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes managed objects used for managing tunnels of any type over IPv4 networks. Extension MIBs may be designed for managing protocol-specific objects. Likewise, extension MIBs may be designed for managing security-specific objects. This MIB does not support tunnels over non-IPv4 networks (including IPv6 networks). Management of such tunnels may be supported by other MIBs.

A similar MIB for tunneling over IPv6 should be defined.

5.73. RFC 2669 DOCSIS Cable Device MIB Cable Device Management Information Base for DOCSIS compliant Cable Modems and Cable Modem Termination Systems

This document states:

> Please note that the DOCSIS 1.0 standard only requires Cable Modems to implement SNMPv1 and to process IPv4 customer traffic. Design choices in this MIB reflect those requirements. Future versions of the DOCSIS standard are expected to require support for SNMPv3 and IPv6 as well.

5.74. RFC 2670 Radio Frequency (RF) Interface Management Information Base for MCNS/DOCSIS compliant RF interfaces

This MIB defines the following objects:

```
DocsIfCmtsCmStatusEntry ::= SEQUENCE {
            docsIfCmtsCmStatusIndex               Integer32,
            docsIfCmtsCmStatusMacAddress          MacAddress,
            docsIfCmtsCmStatusIpAddress           IpAddress,
            docsIfCmtsCmStatusDownChannelIfIndex  InterfaceIndexOrZero,
            docsIfCmtsCmStatusUpChannelIfIndex    InterfaceIndexOrZero,
            docsIfCmtsCmStatusRxPower             TenthdBmV,
            docsIfCmtsCmStatusTimingOffset        Unsigned32,
            docsIfCmtsCmStatusEqualizationData    OCTET STRING,
            docsIfCmtsCmStatusValue               INTEGER,
            docsIfCmtsCmStatusUnerroreds          Counter32,
            docsIfCmtsCmStatusCorrecteds          Counter32,
            docsIfCmtsCmStatusUncorrectables      Counter32,
            docsIfCmtsCmStatusSignalNoise         TenthdB,
            docsIfCmtsCmStatusMicroreflections    Integer32
        }

docsIfCmtsCmStatusIpAddress OBJECT-TYPE
        SYNTAX      IpAddress
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "IP address of this Cable Modem.  If the Cable Modem has no
             IP address assigned, or the IP address is unknown, this
             object returns a value of 0.0.0.0.  If the Cable Modem has
             multiple IP addresses, this object returns the IP address
             associated with the Cable interface."
        ::= { docsIfCmtsCmStatusEntry 3 }
```

This object needs to be deprecated and replaced by one that supports both IPv4 and IPv6 addresses.

5.75. RFC 2674 Definitions of Managed Objects for Bridges with Traffic Classes, Multicast Filtering and Virtual LAN Extensions

There are no IPv4 dependencies in this specification.

5.76. RFC 2677 Definitions of Managed Objects for the NBMA Next Hop Resolution Protocol (NHRP)

There are no IPv4 dependencies in this specification.

5.77. RFC 2720 Traffic Flow Measurement: Meter MIB

This specification is both IPv4 and IPv6 aware and needs no changes.

5.78. RFC 2725 Routing Policy System Security

There are no IPv4 dependencies in this specification.

5.79. RFC 2726 PGP Authentication for RIPE Database Updates

There are no IPv4 dependencies in this specification.

5.80. RFC 2737 Entity MIB (Version 2)

There are no IPv4 dependencies in this specification.

5.81. RFC 2741 Agent Extensibility (AgentX) Protocol Version 1

Although the examples in the document are for IPv4 transport only, there is no IPv4 dependency in the AgentX protocol itself.

5.82. RFC 2742 Definitions of Managed Objects for Extensible SNMP Agents

There are no IPv4 dependencies in this specification.

5.83. RFC 2748 The COPS (Common Open Policy Service) Protocol

This specification is both IPv4 and IPv6 aware and needs no changes.

5.84. RFC 2749 COPS usage for RSVP

There are no IPv4 dependencies in this specification.

5.85. RFC 2769 Routing Policy System Replication

There are no IPv4 dependencies in this specification.

5.86. RFC 2787 Definitions of Managed Objects for the Virtual Router Redundancy Protocol

As stated in the Overview section:

> Since the VRRP protocol is intended for use with IPv4 routers only, this MIB uses the SYNTAX for IP addresses which is specific to IPv4. Thus, changes will be required for this MIB to interoperate in an IPv6 environment.

5.87. RFC 2788 Network Services Monitoring MIB

There are no IPv4 dependencies in this specification.

5.88. RFC 2789 Mail Monitoring MIB

There are no IPv4 dependencies in this specification.

5.89. RFC 2837 Definitions of Managed Objects for the Fabric Element in Fibre Channel Standard

There are no IPv4 dependencies in this specification.

5.90. RFC 2856 Textual Conventions for Additional High Capacity Data Types

There are no IPv4 dependencies in this specification.

5.91. RFC 2864 The Inverted Stack Table Extension to the Interfaces Group MIB

There are no IPv4 dependencies in this specification.

5.92. RFC 2895 Remote Network Monitoring MIB Protocol Identifier Reference

This specification is both IPv4 and IPv6 aware and needs no changes.

5.93. RFC 2925 Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations

This MIB mostly is IPv4 and IPv6 aware. There are a few assumptions that are problems, though. In the following object definitions:

```
pingCtlDataSize OBJECT-TYPE
   SYNTAX      Unsigned32 (0..65507)
   UNITS       "octets"
   MAX-ACCESS  read-create
```

```
      STATUS      current
      DESCRIPTION
          "Specifies the size of the data portion to be
          transmitted in a ping operation in octets.  A ping
          request is usually an ICMP message encoded
          into an IP packet.  An IP packet has a maximum size
          of 65535 octets.  Subtracting the size of the ICMP
          or UDP header (both 8 octets) and the size of the IP
          header (20 octets) yields a maximum size of 65507
          octets."
      DEFVAL { 0 }
      ::= { pingCtlEntry 5 }


   traceRouteCtlDataSize OBJECT-TYPE
      SYNTAX      Unsigned32 (0..65507)
      UNITS       "octets"
      MAX-ACCESS  read-create
      STATUS      current
      DESCRIPTION
          "Specifies the size of the data portion of a traceroute
          request in octets.  A traceroute request is essentially
          transmitted by encoding a UDP datagram into a
          IP packet.  So subtracting the size of a UDP header
          (8 octets) and the size of a IP header (20 octets)
          yields a maximum of 65507 octets."
      DEFVAL { 0 }
      ::= { traceRouteCtlEntry 6 }
```

The DESCRIPTION clauses need to be updated to remove the IPv4 dependencies.

5.94. RFC 2932 IPv4 Multicast Routing MIB

This specification is only defined for IPv4 and a similar MIB must be defined for IPv6.

5.95. RFC 2933 Internet Group Management Protocol MIB

As stated in this document:

> Since IGMP is specific to IPv4, this MIB does not support management of equivalent functionality for other address families, such as IPv6.

5.96. RFC 2940 Definitions of Managed Objects for Common Open Policy Service (COPS) Protocol Clients

This MIB is both IPv4 and IPv6 aware and needs no changes.

5.97. RFC 2954 Definitions of Managed Objects for Frame Relay Service

There are no IPv4 dependencies in this specification.

5.98. RFC 2955 Definitions of Managed Objects for Monitoring and Controlling the Frame Relay/ATM PVC Service Interworking Function

There are no IPv4 dependencies in this specification.

5.99. RFC 2959 Real-Time Transport Protocol Management Information Base

There are no IPv4 dependencies in this specification.

5.100. RFC 2981 Event MIB

There are no IPv4 dependencies in this specification.

5.101. RFC 2982 Distributed Management Expression MIB

There are no IPv4 dependencies in this specification.

5.102. RFC 3014 Notification Log MIB

There are no IPv4 dependencies in this specification.

5.103. RFC 3019 IP Version 6 Management Information Base for The Multicast Listener Discovery Protocol

This is an IPv6 related document and is not discussed in this document.

5.104. RFC 3020 Definitions of Managed Objects for Monitoring and Controlling the UNI/NNI Multilink Frame Relay Function

There are no IPv4 dependencies in this specification.

5.105. RFC 3055 Management Information Base for the PINT Services Architecture

There are no IPv4 dependencies in this specification.

5.106. RFC 3060 Policy Core Information Model -- Version 1 Specification (CIM)

There are no IPv4 dependencies in this specification.

5.107. RFC 3084 COPS Usage for Policy Provisioning (COPS-PR)

This specification builds on RFC 2748, and is both IPv4 and IPv6 capable. The specification defines a sample filter in section 4.3, which has "ipv4" in it.

5.108. RFC 3165 Definitions of Managed Objects for the Delegation of Management Scripts

There are no IPv4 dependencies in this specification.

5.109. RFC 3231 Definitions of Managed Objects for Scheduling Management Operations

There are no IPv4 dependencies in this specification.

5.110. RFC 3291 Textual Conventions for Internet Network Addresses

There are no IPv4 dependencies in this specification.

5.111. RFC 3635 Definitions of Managed Objects for the Ethernet-like Interface Types

There are no IPv4 dependencies in this specification.

5.112. RFC 3636 Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs)

There are no IPv4 dependencies in this specification.

6. Experimental RFCs

Experimental RFCs typically define protocols that do not have widescale implementation or usage on the Internet. They are often propriety in nature or used in limited arenas. They are documented to the Internet community in order to allow potential interoperability or some other potential useful scenario. In a few cases, they are presented as alternatives to the mainstream solution to an acknowledged problem.

6.1. RFC 1187 Bulk Table Retrieval with the SNMP

There are no IPv4 dependencies in this specification.

6.2. RFC 1224 Techniques for managing asynchronously generated alerts

There are no IPv4 dependencies in this specification.

6.3. RFC 1238 CLNS MIB for use with Connectionless Network Protocol (ISO 8473) and End System to Intermediate System (ISO 9542)

There are no IPv4 dependencies in this specification.

6.4. RFC 1592 Simple Network Management Protocol Distributed Protocol Interface Version 2.0

There are no IPv4 dependencies in this specification.

6.5. RFC 1792 TCP/IPX Connection Mib Specification

There are no IPv4 dependencies in this specification.

6.6. RFC 2724 RTFM: New Attributes for Traffic Flow Measurement

There are no IPv4 dependencies in this specification.

6.7. RFC 2758 Definitions of Managed Objects for Service Level Agreements Performance Monitoring

This specification is both IPv4 and IPv6 aware and needs no changes.

6.8. RFC 2786 Diffie-Helman USM Key Management Information Base and Textual Convention

There are no IPv4 dependencies in this specification.

6.9. RFC 2903 Generic AAA Architecture

There are no IPv4 dependencies in this specification.

6.10. RFC 2934 Protocol Independent Multicast MIB for IPv4

This document is specific to IPv4.

6.11. RFC 3179 Script MIB Extensibility Protocol Version 1.1

There are no IPv4 dependencies in this specification.

7. Summary of Results

In the initial survey of RFCs, 36 positives were identified out of a total of 153, broken down as follows:

Standards:	6 out of 15 or 40.00%
Draft Standards:	4 out of 15 or 26.67%
Proposed Standards:	26 out of 112 or 23.21%
Experimental RFCs:	0 out of 11 or 0.00%

Of those identified, many require no action because they document outdated and unused protocols, while others are document protocols that are actively being updated by the appropriate working groups. Additionally there are many instances of standards that should be updated but do not cause any operational impact if they are not updated. The remaining instances are documented below.

7.1. Standards

7.1.1. STD 16, Structure of Management Information (RFCs 1155 and 1212)

RFC 1155 and RFC 1212 (along with the informational document RFC 1215) define SMIv1. These documents have been superseded by RFCs 2578, 2579, and 2580 which define SMIv2. Since SMIv1 is no longer being used as the basis for new IETF MIB modules, the limitations identified in this Internet Standard do not require any action.

7.1.2. STD 17 Simple Network Management Protocol (RFC 1213)

The limitations identified have been addressed, because RFC 1213 has been split into multiple modules which are all IPv6 capable.

7.2. Draft Standards

7.2.1. BGP4 MIB (RFC 1657)

This problem is currently being addressed by the Inter Domain Routing (IDR) WG [2].

7.2.2. SMDS MIB (RFC 1694)

See Internet Area standards. Once a specification for IPv6 over SMDS is created a new MIB must be defined.

7.2.3. RIPv2 MIB (RFC 1724)

There is no updated MIB module to cover the problems outlined. A new MIB module should be defined.

7.2.4. OSPFv2 MIB (RFC 1850)

This problem is currently being addressed by the OSPF WG [3].

7.2.5. Transport MIB (RFC 1906)

RFC 1906 has been obsoleted by RFC 3417, Transport Mappings for SNMP, and the limitations of this specification have been addressed by that RFC, which defines TCs that can be used to specify transport domains in an IP version-independent way. RFC 3419 recommends that those TCs be used in place of SnmpUDPAddress when IPv6 support is required and for all new applications that are not SNMP-specific.

7.3. Proposed Standards

7.3.1. MIB for Multiprotocol Interconnect over X.25 (RFC 1461)

This problem has not been addressed. If a user requirement for IPv6 over X.25 develops (which is thought to be unlikely) then this MIB module will need to be updated in order to accommodate it.

7.3.2. PPP IPCP MIB (RFC 1473)

There is no updated MIB to cover the problems outlined. A new MIB should be defined.

7.3.3. Appletalk MIB (RFC 1742)

This problem has not been addressed. If a user requirement for IPv6 over Appletalk develops (which is thought to be unlikely) then this MIB module will need to be updated (or a new MIB module will need to be created) in order to accommodate it.

7.3.4. The Definitions of Managed Objects for IP Mobility Support using SMIv2 (RFC 2006)

The problems are being resolved by the MIP6 WG [4].

7.3.5. SMIv2 IP MIB (RFC 2011)

This issue is being resolved by the IPv6 WG [5].

7.3.6. SNMPv2 TCP MIB (RFC 2012)

This issue is being resolved by the IPv6 WG [6].

7.3.7. SNMPv2 UDP MIB (RFC 2013)

This issue is being resolved by the IPv6 WG [7].

7.3.8. RMON-II MIB (RFC 2021)

This issue has been brought to the attention of the RMONMIB WG. Currently, there is a work in progress [8] to update RFC 2021, but it does not address the problems that have been identified; it is expected that there will be a resolution in a future version of that document.

7.3.9. DataLink Switching using SMIv2 MIB (RFC 2024)

The problems have not been addressed and an updated MIB should be defined.

7.3.10. IP Forwarding Table MIB (RFC 2096)

This issue is being worked on by the IPv6 WG [9].

7.3.11. Classical IP & ARP over ATM MIB (RFC 2320)

The current version of Classical IP and ARP over ATM (RFC 2225) does not support IPv6. If and when that protocol specification is updated to add IPv6 support, then new MIB objects to represent IPv6 addresses will need to be added to this MIB module.

7.3.12. Multicast over UNI 3.0/3.1 ATM MIB (RFC 2417)

The current version of Multicast over UNI 3.0/3.1 ATM (RFC 2022) does not support IPv6. If and when that protocol specification is updated to add IPv6 support, then new MIB objects to represent IPv6 addresses will need to be added to this MIB module.

7.3.13. ATM MIB (RFC 2515)

The AToM MIB WG is currently collecting implementation reports for RFC 2515 and is considering whether to advance, revise, or retire this specification. The problems identified have been brought to the attention of the WG.

7.3.14. TN3270 MIB (RFC 2562)

The problems identified are not being addressed and a new MIB module may need to be defined.

7.3.15. Application MIB (RFC 2564)

The problems identified are not being addressed and a new MIB module may need to be defined. One possible solution might be to use the RFC 3419 TCs.

7.3.16. Definitions of Managed Objects for APPN/HPR in IP Networks (RFC 2584)

The problems identified are not addressed and a new MIB may be defined.

7.3.17. RADIUS MIB (RFC 2618)

The problems have not been addressed and a new MIB should be defined.

7.3.18. RADIUS Authentication Server MIB (RFC 2619)

The problems have not been addressed and a new MIB should be defined.

7.3.19. RPSL (RFC 2622)

Additional objects must be defined for IPv6 addresses and prefixes.

[10] defines extensions to solve this issue, and it is being considered for publication.

7.3.20. IPv4 Tunnel MIB (RFC 2667)

The issue is being resolved.

7.3.21. DOCSIS MIB (RFC 2669)

This problem is currently being addressed by the IPCDN WG.

7.3.22. RF MIB For DOCSIS (RFC 2670)

This problem is currently being addressed by the IPCDN WG [11].

7.3.23. VRRP MIB (RFC 2787)

The problems have not been addressed and a new MIB may need to be defined.

7.3.24. MIB For Traceroute, Pings and Lookups (RFC 2925)

The problems have not been addressed and a new MIB may need to be defined.

7.3.25. IPv4 Multicast Routing MIB (RFC 2932)

The problems have not been addressed a new MIB must be defined.

7.3.26. IGMP MIB (RFC 2933)

This problem is currently being addressed by the MAGMA WG [12].

7.4. Experimental RFCs

7.4.1. Protocol Independent Multicast MIB for IPv4 (RFC 2934)

The problems have not been addressed and a new MIB may need to be defined.

8. Security Considerations

This memo examines the IPv6-readiness of specifications; this does not have security considerations in itself.

9. Acknowledgements

The authors would like to acknowledge the support of the Internet Society in the research and production of this document. Additionally the author, Philip J. Nesser II, would like to thank his partner in all ways, Wendy M. Nesser.

The editor, Andreas Bergstrom, would like to thank Pekka Savola for his guidance and collection of comments for the editing of this document. He would further like to thank Juergen Schoenwaelder, Brian Carpenter, Bert Wijnen and especially C. M. Heard for feedback on many points of this document.

10. References

10.1. Normative Reference

[1] Nesser, II, P. and A. Bergstrom, Editor, "Introduction to the Survey of IPv4 Addresses in Currently Deployed IETF Standards", RFC 3789, June 2004.

10.2. Informative References

[2] Haas, J. and S. Hares, Editors, "Definitions of Managed Objects for the Fourth Version of Border Gateway Protocol (BGP-4)", Work in Progress, April 2004.

[3] Joyal, D. and V. Manral, "Management Information Base for OSPFv3", Work in Progress, April 2004.

[4] Keeni, G., Koide, K., Nagami, K. and S. Gundavelli, "The Mobile IPv6 MIB", Work in Progress, February 2004.

[5] Routhier, S., Editor, "Management Information Base for the Internet Protocol (IP)", Work in Progress, April 2004.

[6] Raghunarayan, R., Editor, "Management Information Base for the Transmission Control Protocol (TCP)", Work in Progress, February 2004.

[7] Fenner, B. and J. Flick, "Management Information Base for the User Datagram Protocol (UDP)", Work in Progress, April 2004.

[8] Waldbusser, S., "Remote Network Monitoring Management Information Base Version 2 Using SMIv2", Work in Progress, February 2004.

[9] Haberman, B., "IP Forwarding Table MIB", Work in Progress, February 2004.

[10] Blunk, L., Damas, J., Parent, F. and A. Robachevsky, "Routing Policy Specification Language next generation (RPSLng)", Work in Progress, April 2004.

[11] Raftus, D. and E. Cardona, Editor, "Radio Frequency (RF) Interface Management Information Base for DOCSIS 2.0 compliant RF interfaces", Work in Progress, April 2004.

[12] Chesterfield, J., Editor, "Multicast Group Membership Discovery MIB", Work in Progress, February 2004.

11. Authors' Addresses

Please contact the authors with any questions, comments or suggestions at:

Philip J. Nesser II
Principal
Nesser & Nesser Consulting
13501 100th Ave NE, #5202
Kirkland, WA 98034

Phone: +1 425 481 4303
Fax: +1 425 48
EMail: phil@nesser.com

Andreas Bergstrom (Editor)
Ostfold University College
Rute 503 Buer
N-1766 Halden
Norway

EMail: andreas.bergstrom@hiof.no

12. Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.