IPv6 MIB Revision Design Team
INTERNET-DRAFT
Expires: August 2001

Bill Fenner
AT&T Research
Brian Haberman
Nortel Networks
Juergen Schoenwalder
TU Braunschweig
Dave Thaler
Microsoft
February 2001

Management Information Base for the Internet Protocol (IP) draft-ops-rfc2011-update-00.txt

Status of this Document

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.

This document is a product of the IPv6 MIB Revision Design Team. Comments should be addressed to the authors, or the mailing list at ipv6mib@ibr.cs.tu-bs.de.

Copyright Notice

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 managed objects used for implementations of the Internet Protocol (IP) in an IP version independent manner.

Table of Contents

1. The SNMP Management Framework

The SNMP Management Framework presently consists of five major components:

o An overall architecture, described in RFC 2571 [6].

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 [7], STD 16, RFC 1212 [8] and RFC 1215 [9]. The second version, called SMIv2, is described in STD 58, RFC 2578 [10], STD 58, RFC 2579 [11] and STD 58, RFC 2580 [12].

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 [13]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901 [14] and RFC 1906 [15]. The third version of the message protocol is called SNMPv3 and described in RFC 1906 [15], RFC 2572 [16] and RFC 2574 [17].

o Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in

STD 15, RFC 1157 [13]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [18].

o A set of fundamental applications described in RFC 2573 [19] and the view-based access control mechanism described in RFC 2575 [20].

A more detailed introduction to the current SNMP Management Framework can be found in RFC 2570 [21].

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.

2. Revision History

Changes from first draft posted to v6mib mailing list:

23 Feb 2001

Added ipv4InterfaceTable

Added ipv6InterfaceTable

Added ipAddressPrefixTable and slightly reworked ipAddressTable (nee inetAddressTable).

Deprecated ipMIBCompliance. Still need to finish updated ones.

Added copyright and table of contents.

7 Feb 2001

Renamed inetIfStats to ipIfStats

Added ipv6ScopeTable

Added ScopeIdentifier TC, which should be in INET-ADDRESS-MIB.

Added SIZE to inetAddrAddr and inetNetToMediaAddress

Wrote some boilerplate for multi-interface-or-system-wide counter tables.

3. Updating Implementations

Boy, does this need more text. XXX

3.1. Updating an implementation of the IPv4-only IP-MIB

o New counters: InTooBigErrors, InTruncatedPkts, InMcastPkts, OutMcastPkts.

o Can report existing counters as system-wide in ipIfStatsTable (0 for ifindex) or can implement per-interface counters and report those.

o Move ipAdEntReasmMaxSize to ipIfReasmMaxSize

3.2. Updating an implementation of the IPv6-MIB

o New counters: ReasmTimeout

o Don't need to implement per-interface stats.

o Reimplement ipv6IfLowerLayer with the TUNNEL-MIB (note: can't represent IPv6-over-IPv6 since the TUNNEL-MIB is IPv4-only)

4. Definitions

```
IP-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE,
    Integer32, Counter32, IpAddress, mib-2, Unsigned32
                                    FROM SNMPv2-SMI
    PhysAddress, TruthValue, TimeStamp, RowPointer,
    TEXTUAL-CONVENTION -- XXX
                                    FROM SNMPv2-TC
    MODULE-COMPLIANCE, OBJECT-GROUP    FROM SNMPv2-CONF
    InetAddress, InetAddressType,
    InetAddressPrefixLength            FROM INET-ADDRESS-MIB
    InterfaceIndex,
    InterfaceIndexOrZero, ifIndex      FROM IF-MIB;


ipMIB MODULE-IDENTITY
    LAST-UPDATED "200102220000Z"
```

```
    ORGANIZATION "IETF IPv6 MIB Revision Team"
    CONTACT-INFO
           "Editor:
            Bill Fenner
            AT&T Labs - Research
            75 Willow Rd
            Menlo Park, CA

            Phone: +1 650 330-7893
            Email: <fenner@research.att.com>"
    DESCRIPTION
           "The MIB module for managing IP and ICMP implementations, but
            excluding their management of IP routes."
    REVISION      "200102220000Z"
    DESCRIPTION
           "IP version neutral revision, published as RFC XXXX."
    REVISION      "9411010000Z"
    DESCRIPTION
           "Published seperately as RFC 2011."
    REVISION      "9103310000Z"
    DESCRIPTION
           "The initial revision of this MIB module was part of MIB-II."
    ::= { mib-2 48}

-- the IP general group

ip       OBJECT IDENTIFIER ::= { mib-2 4 }

ipForwarding OBJECT-TYPE
    SYNTAX     INTEGER {
                    forwarding(1),    -- acting as a router
                    notForwarding(2)  -- NOT acting as a router
                }
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
           "The indication of whether this entity is acting as an IPv4
            router in respect to the forwarding of datagrams received
            by, but not addressed to, this entity.  IPv4 routers forward
            datagrams.  IPv4 hosts do not (except those source-routed
            via the host)."
    ::= { ip 1 }

ipDefaultTTL OBJECT-TYPE
    SYNTAX     INTEGER (1..255)
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
```

```
           "The default value inserted into the Time-To-Live field of
            the IPv4 header of datagrams originated at this entity,
            whenever a TTL value is not supplied by the transport layer
            protocol."
    ::= { ip 2 }

-- the IPv6 general group

ipv6MIB OBJECT IDENTIFIER    ::= { mib-2 55 }

ipv6MIBObjects OBJECT IDENTIFIER   ::= { ipv6MIB 1 }

ipv6Forwarding OBJECT-TYPE
    SYNTAX     INTEGER {
                          forwarding(1),    -- acting as a router

                                            -- NOT acting as
                          notForwarding(2)  -- a router
                        }
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
           "The indication of whether this entity is acting as an IPv6
            router in respect to the forwarding of datagrams received
            by, but not addressed to, this entity.  IPv6 routers forward
            datagrams.  IPv6 hosts do not (except those source-routed
            via the host).

            Note that for some managed nodes, this object may take on
            only a subset of the values possible.  Accordingly, it is
            appropriate for an agent to return a `wrongValue' response
            if a management station attempts to change this object to an
            inappropriate value."
    ::= { ipv6MIBObjects 1 }

ipv6DefaultHopLimit OBJECT-TYPE
    SYNTAX     INTEGER (0..255)
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
           "The default value inserted into the Hop Limit field of the
            IPv6 header of datagrams originated at this entity, whenever
            a Hop Limit value is not supplied by the transport layer
            protocol."
    ::= { ipv6MIBObjects 2 }

-- XXX what about SIIT object saying whether an IPv4 address
--  describes SIIT mapped or natively mapped on a dual-stack system
```

```
--
-- XXX IP version specific interface tables.
-- XXX This is the part of this new MIB that I'm least sure of.

--
-- IPv4 Interface Table
--

ipv4IfTable OBJECT-TYPE
    SYNTAX     SEQUENCE OF Ipv4IfEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "The table containing per-interface IP-specific information."
    ::= { ip 25 }

ipv4IfEntry OBJECT-TYPE
    SYNTAX     Ipv4IfEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "An entry in the ipv4IfTable."
    INDEX { ipv4IfIndex }
    ::= { ipv4IfTable 1 }

Ipv4IfEntry ::= SEQUENCE {
        ipv4IfIndex         InterfaceIndex,
        ipv4IfReasmMaxSize  Integer32
    }

ipv4IfIndex OBJECT-TYPE
    SYNTAX     InterfaceIndex
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "The interface to which these values apply."
    ::= { ipv4IfEntry 1 }

ipv4IfReasmMaxSize OBJECT-TYPE
    SYNTAX     Integer32 (0..65535)
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The size of the largest IPv4 datagram which this entity can
            re-assemble from incoming IPv4 fragmented datagrams received
            on this interface."
    ::= { ipv4IfEntry 2 }
```

```
--
-- v6 interface table
-- XXX I suspect that most of these objects can go away.
--
-- Open Issues:
--  ipv6InterfaceAdminStatus: does it make sense to enable/disable
--   IPv6 on its own on the interface?
--  ipv6InterfaceOperStatus: other than the above, noIfIdentifier(3)
--   is this one's only useful state, which can be determined from
--   the Address table if DAD failed or there is no v6 address on
--   this interface. [not efficiently, though]

ipv6InterfaceTable OBJECT-TYPE
    SYNTAX     SEQUENCE OF Ipv6InterfaceEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "ipv6 interface table"
    ::= { ip 31 }

ipv6InterfaceEntry OBJECT-TYPE
    SYNTAX     Ipv6InterfaceEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "ipv6 interface entry"
    INDEX { ipv6InterfaceIfIndex }
    ::= { ipv6InterfaceTable 1 }

Ipv6InterfaceEntry ::= SEQUENCE {
        ipv6InterfaceIfIndex           InterfaceIndex,
        ipv6InterfaceEffectiveMtu      Unsigned32,
        ipv6InterfaceReasmMaxSize      Unsigned32,
        ipv6InterfaceIdentifier        Ipv6AddressIfIdentifier,
        ipv6InterfaceIdentifierLength  INTEGER,
        ipv6InterfacePhysicalAddress   PhysAddress
    }

ipv6InterfaceIfIndex OBJECT-TYPE
    SYNTAX     InterfaceIndex
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "The interface."
    ::= { ipv6InterfaceEntry 1 }

ipv6InterfaceEffectiveMtu OBJECT-TYPE
    SYNTAX     Unsigned32
```

```
    UNITS       "octets"
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The size of the largest IPv6 packet which can be
            sent/received on the interface, specified in octets.

            XXX - why isn't this ifMtu - sizeof(ipv6 header)?"
    ::= { ipv6InterfaceEntry 2 }

ipv6InterfaceReasmMaxSize OBJECT-TYPE
    SYNTAX     Unsigned32 (0..65535)
    UNITS       "octets"
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The size of the largest IPv6 datagram which this entity can
            re-assemble from incoming IPv6 fragmented datagrams received
            on this interface."
    ::= { ipv6InterfaceEntry 3 }

-- XXX ugh: I want to get rid of this, which is why it's in the middle
-- of nowhere
Ipv6AddressIfIdentifier ::= TEXTUAL-CONVENTION
     DISPLAY-HINT "2x:"
     STATUS       current
     DESCRIPTION
       "This data type is used to model IPv6 address
       interface identifiers. This is a binary string
        of up to 8 octets in network byte-order."
     SYNTAX      OCTET STRING (SIZE (0..8))

ipv6InterfaceIdentifier OBJECT-TYPE
    SYNTAX     Ipv6AddressIfIdentifier
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
           "The Interface Identifier for this interface that is (at
            least) unique on the link this interface is attached to. The
            Interface Identifier is combined with an address prefix to
            form an interface address.

            By default, the Interface Identifier is autoconfigured
            according to the rules of the link type this interface is
            attached to.

            XXX - is this an EUI64 that belongs more in the IF-MIB?"
    ::= { ipv6InterfaceEntry 4 }
```

```
ipv6InterfaceIdentifierLength OBJECT-TYPE
    SYNTAX     INTEGER (0..64)
    UNITS       "bits"
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
           "The length of the Interface Identifier in bits."
    ::= { ipv6InterfaceEntry 5 }

ipv6InterfacePhysicalAddress OBJECT-TYPE
    SYNTAX     PhysAddress
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The interface's physical address. For example, for an IPv6
            interface attached to an 802.x link, this object normally
            contains a MAC address. Note that in some cases this address
            may differ from the address of the interface's protocol sub-
            layer.  The interface's media-specific MIB must define the
            bit and byte ordering and the format of the value of this
            object. For interfaces which do not have such an address
            (e.g., a serial line), this object should contain an octet
            string of zero length.

            XXX When can this be different from the address of the
            interface's protocol sub-layer, and why?"
    ::= { ipv6InterfaceEntry 6 }




--
-- Per-Interface or System-Wide IP statistics.
--
-- Open issues:
--  Add octet counters similar to ifTable and ifXTable?

ipIfStatsTable OBJECT-TYPE
    SYNTAX     SEQUENCE OF IpIfStatsEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "The table containing traffic statistics.  These statistics
            may be kept per-interface and/or system-wide."
    ::= { ip 26 }

ipIfStatsEntry OBJECT-TYPE
    SYNTAX     IpIfStatsEntry
```

```
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "An interface statistics entry containing objects for a
            particular interface, or system-wide.

            A row with an ipIfStatsIfIndex value of zero indicates a
            system-wide value; a row with a non-zero ipIfStatsIfIndex
            indicates an interface-specific value.  A system may provide
            both system-wide and interface-specific values, in which
            case it is important to note that the system-wide value may
            not be equal to the sum of the interface-specific values
            across all interfaces due to e.g. dynamic interface
            creation/deletion."
    INDEX { ipIfStatsAFType, ipIfStatsIfIndex }
    ::= { ipIfStatsTable 1 }

IpIfStatsEntry ::= SEQUENCE {
        ipIfStatsAFType            InetAddressType,
        ipIfStatsIfIndex           InterfaceIndexOrZero,
        ipIfStatsInReceives        Counter32,
        ipIfStatsInHdrErrors       Counter32,
        ipIfStatsInTooBigErrors    Counter32,
        ipIfStatsInNoRoutes        Counter32,
        ipIfStatsInAddrErrors      Counter32,
        ipIfStatsInUnknownProtos   Counter32,
        ipIfStatsInTruncatedPkts   Counter32,
        ipIfStatsInDiscards        Counter32,
        ipIfStatsInDelivers        Counter32,
        ipIfStatsOutForwDatagrams  Counter32,
        ipIfStatsOutRequests       Counter32,
        ipIfStatsOutDiscards       Counter32,
        ipIfStatsOutFragOKs        Counter32,
        ipIfStatsOutFragFails      Counter32,
        ipIfStatsOutFragCreates    Counter32,
        ipIfStatsReasmReqds        Counter32,
        ipIfStatsReasmOKs          Counter32,
        ipIfStatsReasmFails        Counter32,
        ipIfStatsInMcastPkts       Counter32,
        ipIfStatsOutMcastPkts      Counter32
    }

ipIfStatsAFType OBJECT-TYPE
    SYNTAX     InetAddressType
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "The address family for this row.  May only be IPv4 or IPv6."
```

```
    ::= { ipIfStatsEntry 1 }

ipIfStatsIfIndex OBJECT-TYPE
    SYNTAX     InterfaceIndexOrZero
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "The interface index, or zero for system-wide counters."
    ::= { ipIfStatsEntry 2 }

ipIfStatsInReceives OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The total number of input IP datagrams received by the
            interface, including those received in error."
    ::= { ipIfStatsEntry 3 }

ipIfStatsInHdrErrors OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The number of input IP datagrams discarded due to errors in
            their IP headers, including version number mismatch, other
            format errors, hop count exceeded, errors discovered in
            processing their IP options, etc."
    ::= { ipIfStatsEntry 4 }

ipIfStatsInTooBigErrors OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The number of input IP datagrams that could not be forwarded
            because their size exceeded the link MTU of the outgoing
            interface."
    ::= { ipIfStatsEntry 5 }

ipIfStatsInNoRoutes OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The number of input IP datagrams discarded because no route
            could be found to transmit them to their destination."
    ::= { ipIfStatsEntry 6 }
```

```
ipIfStatsInAddrErrors OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The number of input IP datagrams discarded because the IP
            address in their IP header's destination field was not a
            valid address to be received at this entity.  This count
            includes invalid addresses (e.g., ::0) and unsupported
            addresses (e.g., addresses with unallocated prefixes).  For
            entities which are not IP routers and therefore do not
            forward datagrams, this counter includes datagrams discarded
            because the destination address was not a local address."
    ::= { ipIfStatsEntry 7 }

ipIfStatsInUnknownProtos OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The number of locally-addressed IP datagrams received
            successfully but discarded because of an unknown or
            unsupported protocol. This counter is incremented at the
            interface to which these datagrams were addressed which
            might not be necessarily the input interface for some of the
            datagrams."
    ::= { ipIfStatsEntry 8 }

ipIfStatsInTruncatedPkts OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The number of input IP datagrams discarded because datagram
            frame didn't carry enough data."
    ::= { ipIfStatsEntry 9 }

ipIfStatsInDiscards OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The number of input IP datagrams for which no problems were
            encountered to prevent their continued processing, but which
            were discarded (e.g., for lack of buffer space).  Note that
            this counter does not include any datagrams discarded while
            awaiting re-assembly."
    ::= { ipIfStatsEntry 10 }
```

```
ipIfStatsInDelivers OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The total number of datagrams successfully delivered to IP
            user-protocols (including ICMP).  This counter is
            incremented at the interface to which these datagrams were
            addressed which might not be necessarily the input interface
            for some of the datagrams."
    ::= { ipIfStatsEntry 11 }

ipIfStatsOutForwDatagrams OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The number of output datagrams which this entity received
            and forwarded to their final destinations.  In entities
            which do not act as IP routers, this counter will include
            only those packets which were Source-Routed via this entity,
            and the Source-Route processing was successful.  Note that
            for a successfully forwarded datagram the counter of the
            outgoing interface is incremented."
    ::= { ipIfStatsEntry 12 }

ipIfStatsOutRequests OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The total number of IP datagrams which local IP user-
            protocols (including ICMP) supplied to IP in requests for
            transmission.  Note that this counter does not include any
            datagrams counted in ipIfStatsOutForwDatagrams."
    ::= { ipIfStatsEntry 13 }

ipIfStatsOutDiscards OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The number of output IP datagrams for which no problem was
            encountered to prevent their transmission to their
            destination, but which were discarded (e.g., for lack of
            buffer space).  Note that this counter would include
            datagrams counted in ipIfStatsOutForwDatagrams if any such
            packets met this (discretionary) discard criterion."
```

```
    ::= { ipIfStatsEntry 14 }

ipIfStatsOutFragOKs OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The number of IP datagrams that have been successfully
            fragmented at this output interface."
    ::= { ipIfStatsEntry 15 }

ipIfStatsOutFragFails OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The number of IP datagrams that have been discarded because
            they needed to be fragmented at this output interface but
            could not be."
    ::= { ipIfStatsEntry 16 }

ipIfStatsOutFragCreates OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The number of output datagram fragments that have been
            generated as a result of IP fragmentation at this output
            interface."
    ::= { ipIfStatsEntry 17 }

ipIfStatsReasmReqds OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The number of IP fragments received which needed to be
            reassembled at this interface.  Note that this counter is
            incremented at the interface to which these fragments were
            addressed which might not be necessarily the input interface
            for some of the fragments."
    ::= { ipIfStatsEntry 18 }

ipIfStatsReasmOKs OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
```

```
            "The number of IP datagrams successfully reassembled.  Note
             that this counter is incremented at the interface to which
             these datagrams were addressed which might not be
             necessarily the input interface for some of the fragments."
    ::= { ipIfStatsEntry 19 }

ipIfStatsReasmFails OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
            "The number of failures detected by the IP re-assembly
             algorithm (for whatever reason: timed out, errors, etc.).
             Note that this is not necessarily a count of discarded IP
             fragments since some algorithms (notably the algorithm in
             RFC 815) can lose track of the number of fragments by
             combining them as they are received.  This counter is
             incremented at the interface to which these fragments were
             addressed which might not be necessarily the input interface
             for some of the fragments."
    ::= { ipIfStatsEntry 20 }

ipIfStatsInMcastPkts OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
            "The number of IP multicast packets received by the
             interface"
    ::= { ipIfStatsEntry 21 }

ipIfStatsOutMcastPkts OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
            "The number of IP multicast packets transmitted by the
             interface"
    ::= { ipIfStatsEntry 22 }



--
-- Internet Address Prefix table
--
-- Open Issues:
--  What's OnLinkFlag for IPv4?
--  What's AutonomousFlag for IPv4?
```

```
--  What are PreferredLifetime and ValidLifetime for IPv4?
--  Is there a better SMI data type for *Lifetime objects?

ipAddressPrefixTable OBJECT-TYPE
    SYNTAX     SEQUENCE OF IpAddressPrefixEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "inet prefix table"
    ::= { ip 27 }

ipAddressPrefixEntry OBJECT-TYPE
    SYNTAX     IpAddressPrefixEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "inet prefix entry"
    INDEX    { ipAddressPrefixIfIndex, ipAddressPrefixType,
                                         ipAddressPrefixPrefix,
ipAddressPrefixLength }
    ::= { ipAddressPrefixTable 1 }

IpAddressPrefixEntry ::= SEQUENCE {
        ipAddressPrefixIfIndex              InterfaceIndex,
        ipAddressPrefixType                 InetAddressType,
        ipAddressPrefixPrefix               InetAddress,
        ipAddressPrefixLength               InetAddressPrefixLength,
        ipAddressPrefixOrigin               INTEGER,
        ipAddressPrefixOnLinkFlag           TruthValue,
        ipAddressPrefixAutonomousFlag       TruthValue,
        ipAddressPrefixAdvPreferredLifetime  Unsigned32,
        ipAddressPrefixAdvValidLifetime     Unsigned32
    }

ipAddressPrefixIfIndex OBJECT-TYPE
    SYNTAX     InterfaceIndex
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "The interface on which this prefix is configured."
    ::= { ipAddressPrefixEntry 1 }

ipAddressPrefixType OBJECT-TYPE
    SYNTAX     InetAddressType
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "The address type of ipAddressPrefix.  Only IPv4 and IPv6
```

addresses are expected."

```
    ::= { ipAddressPrefixEntry 2 }

ipAddressPrefixPrefix OBJECT-TYPE
    SYNTAX     InetAddress (SIZE(0..36))
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "The address prefix.  Bits after ipAddressPrefixLength must
            be zero."
    ::= { ipAddressPrefixEntry 3 }

ipAddressPrefixLength OBJECT-TYPE
    SYNTAX     InetAddressPrefixLength
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "The prefix length associated with this prefix."
    ::= { ipAddressPrefixEntry 4 }

ipAddressPrefixOrigin OBJECT-TYPE
    SYNTAX     INTEGER {
        other(1),
                                  manual(2),
                                  wellknown(3),
                                  dhcp(4),
                                  routeradv(5)
    }
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The origin of this prefix.  manual(2) indicates a prefix
            that was manually configured.  wellknown(3) indicates a
            well-known prefix, e.g. xxx.yyy/16 for IPv4
            autoconfiguration.  dhcp(4) indicates a prefix that was
            assigned by a DHCP server.  routeradv(5) indicates a prefix
            learned from a router advertisement."
    ::= { ipAddressPrefixEntry 5 }

ipAddressPrefixOnLinkFlag OBJECT-TYPE
    SYNTAX     TruthValue
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "This object has the value 'true(1)', if this prefix can be
            used  for on-link determination and the value 'false(2)'
            otherwise."
    ::= { ipAddressPrefixEntry 6 }
```

```
ipAddressPrefixAutonomousFlag OBJECT-TYPE
    SYNTAX     TruthValue
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "Autonomous address configuration flag. When true(1),
            indicates that this prefix can be used for autonomous
            address configuration (i.e. can be used to form a local
            interface address).  If false(2), it is not used to
            autoconfigure a local interface address."
    ::= { ipAddressPrefixEntry 7 }

ipAddressPrefixAdvPreferredLifetime OBJECT-TYPE
    SYNTAX     Unsigned32
    UNITS       "seconds"
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The length of time in seconds that this prefix will remain
            preferred, i.e. time until deprecation.  A value of
            4,294,967,295 represents infinity.

            The address generated from a deprecated prefix should no
            longer be used as a source address in new communications,
            but packets received on such an interface are processed as
            expected."
    ::= { ipAddressPrefixEntry 8 }

ipAddressPrefixAdvValidLifetime OBJECT-TYPE
    SYNTAX     Unsigned32
    UNITS       "seconds"
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The length of time in seconds that this prefix will remain
            valid, i.e. time until invalidation.  A value of
            4,294,967,295 represents infinity.

            The address generated from an invalidated prefix should not
            appear as the destination or source address of a packet."
    ::= { ipAddressPrefixEntry 9 }




--
-- Internet Address Table
--
```

```
-- Open Issues:
--  should ipAddressv4BcastAddr go somewhere else?
--  meeting notes said: dave: pointer to prefix table.  What's that mean?

ipAddressTable OBJECT-TYPE
    SYNTAX     SEQUENCE OF IpAddressEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "inet addr table"
    ::= { ip 28 }

ipAddressEntry OBJECT-TYPE
    SYNTAX     IpAddressEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "inet addr entry"
    INDEX { ipAddressAddrType, ipAddressAddr }
    ::= { ipAddressTable 1 }

IpAddressEntry ::= SEQUENCE {
        ipAddressAddrType  InetAddressType,
        ipAddressAddr      InetAddress,
        ipAddressIfIndex   InterfaceIndex,
        ipAddressType      INTEGER,
        ipAddressPrefix    RowPointer,
        ipAddressOrigin    INTEGER,
        ipAddressStatus    INTEGER
    }

ipAddressAddrType OBJECT-TYPE
    SYNTAX     InetAddressType
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "The address type of ipAddressAddr."
    ::= { ipAddressEntry 1 }

ipAddressAddr OBJECT-TYPE
    SYNTAX     InetAddress (SIZE(0..36))
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "The IP address to which this entry's addressing information
            pertains."
    ::= { ipAddressEntry 2 }
```

```
ipAddressIfIndex OBJECT-TYPE
    SYNTAX     InterfaceIndex
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The index value which uniquely identifies the interface to
            which this entry is applicable.  The interface identified by
            a particular value of this index is the same interface as
            identified by the same value of RFC 2863's ifIndex."
    ::= { ipAddressEntry 3 }

ipAddressType OBJECT-TYPE
    SYNTAX     INTEGER {
                 unicast(1),
                 anycast(2),
                 broadcast(3)
    }
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The type of address."
    ::= { ipAddressEntry 4 }

ipAddressPrefix OBJECT-TYPE
    SYNTAX     RowPointer
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "A pointer to the row in the prefix table to which this
            address belongs.  May be { 0 0 } if there is no such row."
    ::= { ipAddressEntry 5 }

ipAddressOrigin OBJECT-TYPE
    SYNTAX     INTEGER {
                 other(1),
                                              manual(2),
                                              wellknown(3),
                                              dhcp(4),-- XXX or
assignedbyserver ?
                                              linklayer(5),
                                              random(6)
    }
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The origin of the address.  manual(2) indicates that the
            address was manually configured.  wellknown(3) indicates an
            address constructed from a well-known value, e.g. an IANA-
```

assigned anycast address. dhcp(4) indicates an address that

```
            was assigned to this system by a DHCP server.  linklayer(5)
            indicates an address created by IPv6 stateless
            autoconfiguration.  random(6) indicates an address chosen by
            random, e.g. an IPv4 address within xxx.yyy/16."
    ::= { ipAddressEntry 6 }

ipAddressStatus OBJECT-TYPE
    SYNTAX     INTEGER {
                     preferred(1),

                     deprecated(2),

                     invalid(3),

                     inaccessible(4),

                     unknown(5),  -- status can not be determined
                                  -- for some reason.
                                          tentative(6),

                                          duplicate(7)
                    }
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "Address status.  The preferred(1) state indicates that this
            is a valid address that can appear as the destination or
            source address of a packet.  The deprecated(2) state
            indicates that this is a valid but deprecated address that
            should no longer be used as a source address in new
            communications, but packets addressed to such an address are
            processed as expected. The invalid(3) state indicates that
            this is not valid address which should not appear as the
            destination or source address of a packet. The
            inaccessible(4) state indicates that the address is not
            accessible because the interface to which this address is
            assigned is not operational.  The tentative(6) state
            indicates the uniqueness of the address on the link is being
            verified.  The duplicate(7) state indicates the address has
            been determined to be non-unique on the link and so must not
            be used.

            In the absence of other information, an IPv4 address is
            always preferred(1)."
    ::= { ipAddressEntry 7 }
```

```
-- the Internet Address Translation table

-- The Address Translation tables contain the IP address to
-- "physical" address equivalences.  Some interfaces do not
-- use translation tables for determining address
-- equivalences (e.g., DDN-X.25 has an algorithmic method);
-- if all interfaces are of this type, then the Address
-- Translation table is empty, i.e., has zero entries.

--
-- Open issues:
--  inetNetToMediaState - what values for !ipv6?  noNUD(7) or unknown(6)?
--  inetNetToMediaState - why no value for incomplete?

inetNetToMediaTable OBJECT-TYPE
    SYNTAX     SEQUENCE OF InetNetToMediaEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "The IP Address Translation table used for mapping from IP
            addresses to physical addresses.

            The Address Translation tables contain the IP address to
            'physical' address equivalences.  Some interfaces do not use
            translation tables for determining address equivalences
            (e.g., DDN-X.25 has an algorithmic method); if all
            interfaces are of this type, then the Address Translation
            table is empty, i.e., has zero entries."
    ::= { ip 29 }

inetNetToMediaEntry OBJECT-TYPE
    SYNTAX     InetNetToMediaEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "Each entry contains one IP address to `physical' address
            equivalence."
    INDEX       { ifIndex,
                  inetNetToMediaNetAddressType,
                  inetNetToMediaNetAddress }
    ::= { inetNetToMediaTable 1 }

InetNetToMediaEntry ::= SEQUENCE {
        inetNetToMediaNetAddressType  InetAddressType,
        inetNetToMediaNetAddress      InetAddress,
        inetNetToMediaPhysAddress     PhysAddress,
        inetNetToMediaLastUpdated     TimeStamp,
        inetNetToMediaType            INTEGER,
```

```
        inetNetToMediaState           INTEGER
    }

inetNetToMediaNetAddressType OBJECT-TYPE
    SYNTAX     InetAddressType
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "The type of inetNetToMediaNetAddress."
    ::= { inetNetToMediaEntry 1 }

inetNetToMediaNetAddress OBJECT-TYPE
    SYNTAX     InetAddress (SIZE(0..36))
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "The IP Address corresponding to the media-dependent
            `physical' address."
    ::= { inetNetToMediaEntry 2 }

inetNetToMediaPhysAddress OBJECT-TYPE
    SYNTAX     PhysAddress
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
           "The media-dependent `physical' address."
    ::= { inetNetToMediaEntry 3 }

inetNetToMediaLastUpdated OBJECT-TYPE
    SYNTAX     TimeStamp
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The value of sysUpTime at the time this entry was last
            updated.  If this entry was updated prior to the last re-
            initialization of the local network management subsystem,
            then this object contains a zero value."
    ::= { inetNetToMediaEntry 4 }

inetNetToMediaType OBJECT-TYPE
    SYNTAX     INTEGER {
                other(1),        -- none of the following
                invalid(2),      -- an invalidated mapping
                dynamic(3),
                static(4),
                local(5)         -- local interface
            }
    MAX-ACCESS read-create
```

```
    STATUS     current
    DESCRIPTION
           "The type of mapping.

            Setting this object to the value invalid(2) has the effect
            of invalidating the corresponding entry in the
            inetNetToMediaTable.  That is, it effectively disassociates
            the interface identified with said entry from the mapping
            identified with said entry.  It is an implementation-
            specific matter as to whether the agent removes an
            invalidated entry from the table.  Accordingly, management
            stations must be prepared to receive tabular information
            from agents that corresponds to entries not currently in
            use.  Proper interpretation of such entries requires
            examination of the relevant inetNetToMediaType object.

            The 'dynamic(3)' type indicates that the IP address to
            physical addresses mapping has been dynamically resolved
            using e.g. IPv4 ARP or the IPv6 Neighbor Discovery protocol.
            The 'static(4)' type indicates that the mapping has been
            statically configured.  The 'local(5)' type indicates that
            the mapping is provided for an entity's own interface
            address."
    ::= { inetNetToMediaEntry 5 }

inetNetToMediaState OBJECT-TYPE
    SYNTAX     INTEGER {
                     reachable(1), -- confirmed reachability

                     stale(2),     -- unconfirmed reachability

                     delay(3),     -- waiting for reachability
                                   -- confirmation before entering
                                   -- the probe state

                     probe(4),     -- actively probing

                     invalid(5),   -- an invalidated mapping

                     unknown(6),   -- state can not be determined
                                   -- for some reason.
                               incomplete(7) -- address resolution is being
performed.
                   }
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The Neighbor Unreachability Detection [3] state for the
            interface when the address mapping in this entry is used.
```

```
            If Neighbor Unreachability Detection is not in use (e.g. for
            IPv4), this object is always unknown(6) XXX ?noNUD(8)?."
    REFERENCE "RFC2461"
    ::= { inetNetToMediaEntry 6 }



--
-- The IPv6 Scope Identifier Table.
--
-- Open Issues:
--   Should there be associated objects to provide a scope description,
--   similar to ipMRouteScopeNameString?

-- XXX ScopeIdentifier TC should move to INET-ADDRESS-MIB
ScopeIdentifier ::= TEXTUAL-CONVENTION
    STATUS     current
    DESCRIPTION
           "A Scope Identifier identifies an instance of a specific
            scope.

            The scope identifier MUST disambiguate identical address
            values. For link-local addresses, the scope identifier will
            typically be the interface index (ifIndex as defined in the
            IF-MIB, RFC 2233) of the interface on which the address is
            configured.

            The scope identifier may contain the special value 0 which
            refers to the default scope. The default scope may be used
            in cases where the valid scope identifier is not known
            (e.g., a management application needs to write a site-local
            InetAddressIPv6 address without knowing the site identifier
            value). The default scope SHOULD NOT be used as an easy way
            out in cases where the scope identifier for a non-global
            IPv6 address is known."
    SYNTAX     Unsigned32

ipv6ScopeIdTable OBJECT-TYPE
    SYNTAX     SEQUENCE OF Ipv6ScopeIdEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "The table used to describe IPv6 unicast and multicast scope
            zones."
    ::= { ip 30 }

ipv6ScopeIdEntry OBJECT-TYPE
    SYNTAX     Ipv6ScopeIdEntry
```

```
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "Each entry contains the list of scope identifiers on a given
            interface."
    INDEX { ipv6ScopeIdIfIndex }
    ::= { ipv6ScopeIdTable 1 }

Ipv6ScopeIdEntry ::= SEQUENCE {
        ipv6ScopeIdIfIndex            InterfaceIndex,
        ipv6ScopeIdLinkLocal          ScopeIdentifier,
        ipv6ScopeIdSubnetLocal        ScopeIdentifier,
        ipv6ScopeIdAdminLocal         ScopeIdentifier,
        ipv6ScopeIdSiteLocal          ScopeIdentifier,
        ipv6ScopeId6                  ScopeIdentifier,
        ipv6ScopeId7                  ScopeIdentifier,
        ipv6ScopeIdOrganizationLocal  ScopeIdentifier,
        ipv6ScopeId9                  ScopeIdentifier,
        ipv6ScopeIdA                  ScopeIdentifier,
        ipv6ScopeIdB                  ScopeIdentifier,
        ipv6ScopeIdC                  ScopeIdentifier,
        ipv6ScopeIdD                  ScopeIdentifier
    }

ipv6ScopeIdIfIndex OBJECT-TYPE
    SYNTAX     InterfaceIndex
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "The interface to which these scopes belong."
    ::= { ipv6ScopeIdEntry 1 }

ipv6ScopeIdLinkLocal OBJECT-TYPE
    SYNTAX     ScopeIdentifier
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The Scope Identifier for the link-local scope on this
            interface."
    ::= { ipv6ScopeIdEntry 2 }

ipv6ScopeIdSubnetLocal OBJECT-TYPE
    SYNTAX     ScopeIdentifier
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The Scope Identifier for the subnet-local scope on this
            interface."
```

```
    ::= { ipv6ScopeIdEntry 3 }

ipv6ScopeIdAdminLocal OBJECT-TYPE
    SYNTAX     ScopeIdentifier
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The Scope Identifier for the admin-local scope on this
            interface."
    ::= { ipv6ScopeIdEntry 4 }

ipv6ScopeIdSiteLocal OBJECT-TYPE
    SYNTAX     ScopeIdentifier
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The Scope Identifier for the site-local scope on this
            interface."
    ::= { ipv6ScopeIdEntry 5 }

ipv6ScopeId6 OBJECT-TYPE
    SYNTAX     ScopeIdentifier
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The Scope Identifier for scope 6 on this interface."
    ::= { ipv6ScopeIdEntry 6 }

ipv6ScopeId7 OBJECT-TYPE
    SYNTAX     ScopeIdentifier
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The Scope Identifier for scope 7 on this interface."
    ::= { ipv6ScopeIdEntry 7 }

ipv6ScopeIdOrganizationLocal OBJECT-TYPE
    SYNTAX     ScopeIdentifier
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The Scope Identifier for the orgainzation-local scope on
            this interface."
    ::= { ipv6ScopeIdEntry 8 }

ipv6ScopeId9 OBJECT-TYPE
    SYNTAX     ScopeIdentifier
    MAX-ACCESS read-only
```

```
    STATUS     current
    DESCRIPTION
           "The Scope Identifier for scope 9 on this interface."
    ::= { ipv6ScopeIdEntry 9 }

ipv6ScopeIdA OBJECT-TYPE
    SYNTAX     ScopeIdentifier
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The Scope Identifier for scope A on this interface."
    ::= { ipv6ScopeIdEntry 10 }

ipv6ScopeIdB OBJECT-TYPE
    SYNTAX     ScopeIdentifier
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The Scope Identifier for scope B on this interface."
    ::= { ipv6ScopeIdEntry 11 }

ipv6ScopeIdC OBJECT-TYPE
    SYNTAX     ScopeIdentifier
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The Scope Identifier for scope C on this interface."
    ::= { ipv6ScopeIdEntry 12 }

ipv6ScopeIdD OBJECT-TYPE
    SYNTAX     ScopeIdentifier
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The Scope Identifier for scope D on this interface."
    ::= { ipv6ScopeIdEntry 13 }



icmp     OBJECT IDENTIFIER ::= { mib-2 5 }

--
-- ICMP non-message-specific counters
--
--
-- To do:
--  expand table DESCRIPTION to describe index
--   (including whether an agent MUST support system-wide, per-if,
```

```
--    both, or neither, to be compliant to this MIB.
--    Also, it might be useful to remind readers that the
--    system-wide value is not the sum of the per-if counters.)
-- ************************************************************

inetIcmpTable OBJECT-TYPE
    SYNTAX     SEQUENCE OF InetIcmpEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "The table of generic ICMP counters.  These counters may be
            kept per-interface and/or system-wide."
    ::= { icmp 27 }

inetIcmpEntry OBJECT-TYPE
    SYNTAX     InetIcmpEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "A conceptual row in the inetIcmpTable.

            A row with an inetIcmpIfIndex value of zero indicates a
            system-wide value; a row with a non-zero inetIcmpIfIndex
            indicates an interface-specific value.  A system may provide
            both system-wide and interface-specific values, in which
            case it is important to note that the system-wide value may
            not be equal to the sum of the interface-specific values
            across all interfaces due to e.g. dynamic interface
            creation/deletion."
    INDEX    { inetIcmpAFType, inetIcmpIfIndex }
    ::= { inetIcmpTable 1 }

InetIcmpEntry ::= SEQUENCE {
        inetIcmpAFType     InetAddressType,
        inetIcmpIfIndex    InterfaceIndexOrZero,
        inetIcmpInMsgs     Counter32,
        inetIcmpInErrors   Counter32,
        inetIcmpOutMsgs    Counter32,
        inetIcmpOutErrors  Counter32
    }

inetIcmpAFType OBJECT-TYPE
    SYNTAX     InetAddressType
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "The IP address family of the statistics."
    ::= { inetIcmpEntry 1 }
```

```
inetIcmpIfIndex OBJECT-TYPE
    SYNTAX     InterfaceIndexOrZero
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "The ifindex of the interface, or zero for system-wide
            stats."
    ::= { inetIcmpEntry 2 }

inetIcmpInMsgs OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The total number of ICMP messages which the entity received.
            Note that this counter includes all those counted by
            inetIcmpInErrors."
    ::= { inetIcmpEntry 3 }

inetIcmpInErrors OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The number of ICMP messages which the entity received but
            determined as having ICMP-specific errors (bad ICMP
            checksums, bad length, etc.)."
    ::= { inetIcmpEntry 4 }

inetIcmpOutMsgs OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The total number of ICMP messages which the entity received.
            Note that this counter includes all those counted by
            inetIcmpOutErrors."
    ::= { inetIcmpEntry 5 }

inetIcmpOutErrors OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The number of ICMP messages which this entity did not send
            due to problems discovered within ICMP such as a lack of
            buffers.  This value should not include errors discovered
            outside the ICMP layer such as the inability of IP to route
```

```
            the resultant datagram.  In some implementations there may
            be no types of error which contribute to this counter's
            value."
    ::= { inetIcmpEntry 6 }



--
-- per-AF, per-interface(optionally), per-msg type and code ICMP counters
--


inetIcmpMsgTable OBJECT-TYPE
    SYNTAX     SEQUENCE OF InetIcmpMsgEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "The table of per-message ICMP counters.  These counters may
            be kept per-interface and/or system-wide."
    ::= { icmp 28 }

inetIcmpMsgEntry OBJECT-TYPE
    SYNTAX     InetIcmpMsgEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "A conceptual row in the inetIcmpMsgTable.

            A row with an inetIcmpMsgIfIndex value of zero indicates a
            system-wide value; a row with a non-zero inetIcmpMsgIfIndex
            indicates an interface-specific value.  A system may provide
            both system-wide and interface-specific values, in which
            case it is important to note that the system-wide value may
            not be equal to the sum of the interface-specific values
            across all interfaces due to e.g. dynamic interface
            creation/deletion.

            XXX How to phrase this if?  If the system keeps track of
            individual ICMP code values (e.g.  destination unreachable,
            code administratively prohibited), it creates several rows
            for each inetIcmpMsgType, each with an appropriate value of
            inetIcmpMsgCode.  A row with the special value of
            inetIcmpMsgCode, 256, counts all packets with type
            inetIcmpMsgType that aren't counted in rows with a value of
            inetIcmpMsgCode other than 256."
    INDEX { inetIcmpMsgAFType, inetIcmpMsgIfIndex, inetIcmpMsgType,
            inetIcmpMsgCode }
    ::= { inetIcmpMsgTable 1 }
```

```
InetIcmpMsgEntry ::= SEQUENCE {
        inetIcmpMsgAFType   InetAddressType,
        inetIcmpMsgIfIndex  InterfaceIndexOrZero,
        inetIcmpMsgType     Integer32,
        inetIcmpMsgCode     Integer32,
        inetIcmpMsgInPkts   Counter32,
        inetIcmpMsgOutPkts  Counter32
    }

inetIcmpMsgAFType OBJECT-TYPE
    SYNTAX     InetAddressType
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "The IP address family of the statistics."
    ::= { inetIcmpMsgEntry 1 }

inetIcmpMsgIfIndex OBJECT-TYPE
    SYNTAX     InterfaceIndexOrZero
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "The ifindex of the interface, or zero for system-wide
            stats."
    ::= { inetIcmpMsgEntry 2 }

inetIcmpMsgType OBJECT-TYPE
    SYNTAX     Integer32 (0..255)
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "The ICMP type field of the message type being counted by
            this row."
    ::= { inetIcmpMsgEntry 3 }

inetIcmpMsgCode OBJECT-TYPE
    SYNTAX     Integer32 (0..256)
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "The ICMP code field of the message type being counted by
            this row, or the special value 256 if no specific ICMP code
            is counted by this row."
    ::= { inetIcmpMsgEntry 4 }

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

```
    STATUS     current
    DESCRIPTION
           "The number of input packets for this AF, ifindex, type,
            code."
    ::= { inetIcmpMsgEntry 5 }

inetIcmpMsgOutPkts OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The number of output packets for this AF, ifindex, type,
            code."
    ::= { inetIcmpMsgEntry 6 }



-- XXX
-- To do: move current conformance information here.

--
-- Deprecated objects
--

ipInReceives OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The total number of input datagrams received from
            interfaces, including those received in error."
    ::= { ip 3 }

ipInHdrErrors OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of input datagrams discarded due to errors in
            their IP headers, including bad checksums, version number
            mismatch, other format errors, time-to-live exceeded, errors
            discovered in processing their IP options, etc."
    ::= { ip 4 }

ipInAddrErrors OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
```

```
    DESCRIPTION
           "The number of input datagrams discarded because the IP
            address in their IP header's destination field was not a
            valid address to be received at this entity.  This count
            includes invalid addresses (e.g., 0.0.0.0) and addresses of
            unsupported Classes (e.g., Class E).  For entities which are
            not IP routers and therefore do not forward datagrams, this
            counter includes datagrams discarded because the destination
            address was not a local address."
    ::= { ip 5 }

ipForwDatagrams OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of input datagrams for which this entity was not
            their final IP destination, as a result of which an attempt
            was made to find a route to forward them to that final
            destination.  In entities which do not act as IP routers,
            this counter will include only those packets which were
            Source-Routed via this entity, and the Source-Route option
            processing was successful."
    ::= { ip 6 }

ipInUnknownProtos OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of locally-addressed datagrams received
            successfully but discarded because of an unknown or
            unsupported protocol."
    ::= { ip 7 }

ipInDiscards OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of input IP datagrams for which no problems were
            encountered to prevent their continued processing, but which
            were discarded (e.g., for lack of buffer space).  Note that
            this counter does not include any datagrams discarded while
            awaiting re-assembly."
    ::= { ip 8 }

ipInDelivers OBJECT-TYPE
```

```
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The total number of input datagrams successfully delivered
            to IP user-protocols (including ICMP)."
    ::= { ip 9 }

ipOutRequests OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The total number of IP datagrams which local IP user-
            protocols (including ICMP) supplied to IP in requests for
            transmission.  Note that this counter does not include any
            datagrams counted in ipForwDatagrams."
    ::= { ip 10 }

ipOutDiscards OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of output IP datagrams for which no problem was
            encountered to prevent their transmission to their
            destination, but which were discarded (e.g., for lack of
            buffer space).  Note that this counter would include
            datagrams counted in ipForwDatagrams if any such packets met
            this (discretionary) discard criterion."
    ::= { ip 11 }

ipOutNoRoutes OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of IP datagrams discarded because no route could
            be found to transmit them to their destination.  Note that
            this counter includes any packets counted in ipForwDatagrams
            which meet this `no-route' criterion.  Note that this
            includes any datagrams which a host cannot route because all
            of its default routers are down."
    ::= { ip 12 }

ipReasmTimeout OBJECT-TYPE
    SYNTAX     Integer32
    MAX-ACCESS read-only
```

```
    STATUS     deprecated
    DESCRIPTION
           "The maximum number of seconds which received fragments are
            held while they are awaiting reassembly at this entity."
    ::= { ip 13 }

ipReasmReqds OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of IP fragments received which needed to be
            reassembled at this entity."
    ::= { ip 14 }

ipReasmOKs OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of IP datagrams successfully re-assembled."
    ::= { ip 15 }

ipReasmFails OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of failures detected by the IP re-assembly
            algorithm (for whatever reason: timed out, errors, etc).
            Note that this is not necessarily a count of discarded IP
            fragments since some algorithms (notably the algorithm in
            RFC 815) can lose track of the number of fragments by
            combining them as they are received."
    ::= { ip 16 }

ipFragOKs OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of IP datagrams that have been successfully
            fragmented at this entity."
    ::= { ip 17 }

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

```
    STATUS     deprecated
    DESCRIPTION
           "The number of IP datagrams that have been discarded because
            they needed to be fragmented at this entity but could not
            be, e.g., because their Don't Fragment flag was set."
    ::= { ip 18 }

ipFragCreates OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of IP datagram fragments that have been generated
            as a result of fragmentation at this entity."
    ::= { ip 19 }

ipRoutingDiscards OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of routing entries which were chosen to be
            discarded even though they are valid.  One possible reason
            for discarding such an entry could be to free-up buffer
            space for other routing entries."
    ::= { ip 23 }

-- the deprecated IP address table

ipAddrTable OBJECT-TYPE
    SYNTAX     SEQUENCE OF IpAddrEntry
    MAX-ACCESS not-accessible
    STATUS     deprecated
    DESCRIPTION
           "The table of addressing information relevant to this
            entity's IP addresses."
    ::= { ip 20 }

ipAddrEntry OBJECT-TYPE
    SYNTAX     IpAddrEntry
    MAX-ACCESS not-accessible
    STATUS     deprecated
    DESCRIPTION
           "The addressing information for one of this entity's IP
            addresses."
    INDEX      { ipAdEntAddr }
    ::= { ipAddrTable 1 }
```

```
IpAddrEntry ::= SEQUENCE {
        ipAdEntAddr          IpAddress,
        ipAdEntIfIndex       INTEGER,
        ipAdEntNetMask       IpAddress,
        ipAdEntBcastAddr     INTEGER,
        ipAdEntReasmMaxSize  INTEGER
    }


ipAdEntAddr OBJECT-TYPE
    SYNTAX     IpAddress
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The IP address to which this entry's addressing information
            pertains."
    ::= { ipAddrEntry 1 }

ipAdEntIfIndex OBJECT-TYPE
    SYNTAX     INTEGER (1..2147483647)
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The index value which uniquely identifies the interface to
            which this entry is applicable.  The interface identified by
            a particular value of this index is the same interface as
            identified by the same value of RFC 2863's ifIndex."
    ::= { ipAddrEntry 2 }

ipAdEntNetMask OBJECT-TYPE
    SYNTAX     IpAddress
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The subnet mask associated with the IP address of this
            entry.  The value of the mask is an IP address with all the
            network bits set to 1 and all the hosts bits set to 0."
    ::= { ipAddrEntry 3 }

ipAdEntBcastAddr OBJECT-TYPE
    SYNTAX     INTEGER (0..1)
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The value of the least-significant bit in the IP broadcast
            address used for sending datagrams on the (logical)
            interface associated with the IP address of this entry.  For
            example, when the Internet standard all-ones broadcast
```

```
            address is used, the value will be 1.  This value applies to
            both the subnet and network broadcasts addresses used by the
            entity on this (logical) interface."
    ::= { ipAddrEntry 4 }

ipAdEntReasmMaxSize OBJECT-TYPE
    SYNTAX     INTEGER (0..65535)
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The size of the largest IP datagram which this entity can
            re-assemble from incoming IP fragmented datagrams received
            on this interface."
    ::= { ipAddrEntry 5 }




-- the deprecated IP Address Translation table

-- The Address Translation tables contain the IpAddress to
-- "physical" address equivalences.  Some interfaces do not
-- use translation tables for determining address
-- equivalences (e.g., DDN-X.25 has an algorithmic method);
-- if all interfaces are of this type, then the Address
-- Translation table is empty, i.e., has zero entries.

ipNetToMediaTable OBJECT-TYPE
    SYNTAX     SEQUENCE OF IpNetToMediaEntry
    MAX-ACCESS not-accessible
    STATUS     deprecated
    DESCRIPTION
           "The IP Address Translation table used for mapping from IP
            addresses to physical addresses."
    ::= { ip 22 }

ipNetToMediaEntry OBJECT-TYPE
    SYNTAX     IpNetToMediaEntry
    MAX-ACCESS not-accessible
    STATUS     deprecated
    DESCRIPTION
           "Each entry contains one IpAddress to `physical' address
            equivalence."
    INDEX       { ipNetToMediaIfIndex,
                  ipNetToMediaNetAddress }
    ::= { ipNetToMediaTable 1 }

IpNetToMediaEntry ::= SEQUENCE {
```

```
        ipNetToMediaIfIndex      INTEGER,
        ipNetToMediaPhysAddress  PhysAddress,
        ipNetToMediaNetAddress   IpAddress,
        ipNetToMediaType         INTEGER
    }

ipNetToMediaIfIndex OBJECT-TYPE
    SYNTAX     INTEGER (1..2147483647)
    MAX-ACCESS read-create
    STATUS     deprecated
    DESCRIPTION
           "The interface on which this entry's equivalence is
            effective.  The interface identified by a particular value
            of this index is the same interface as identified by the
            same value of RFC 2863's ifIndex."
    ::= { ipNetToMediaEntry 1 }

ipNetToMediaPhysAddress OBJECT-TYPE
    SYNTAX     PhysAddress
    MAX-ACCESS read-create
    STATUS     deprecated
    DESCRIPTION
           "The media-dependent `physical' address."
    ::= { ipNetToMediaEntry 2 }

ipNetToMediaNetAddress OBJECT-TYPE
    SYNTAX     IpAddress
    MAX-ACCESS read-create
    STATUS     deprecated
    DESCRIPTION
           "The IpAddress corresponding to the media-dependent
            `physical' address."
    ::= { ipNetToMediaEntry 3 }

ipNetToMediaType OBJECT-TYPE
    SYNTAX     INTEGER {
                other(1),        -- none of the following
                invalid(2),      -- an invalidated mapping
                dynamic(3),
                static(4)
            }
    MAX-ACCESS read-create
    STATUS     deprecated
    DESCRIPTION
           "The type of mapping.

            Setting this object to the value invalid(2) has the effect
            of invalidating the corresponding entry in the
```

```
            ipNetToMediaTable.  That is, it effectively disassociates
            the interface identified with said entry from the mapping
            identified with said entry.  It is an implementation-
            specific matter as to whether the agent removes an
            invalidated entry from the table.  Accordingly, management
            stations must be prepared to receive tabular information
            from agents that corresponds to entries not currently in
            use.  Proper interpretation of such entries requires
            examination of the relevant ipNetToMediaType object."
    ::= { ipNetToMediaEntry 4 }



-- the deprecated ICMP group

icmpInMsgs OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The total number of ICMP messages which the entity received.
            Note that this counter includes all those counted by
            icmpInErrors."
    ::= { icmp 1 }

icmpInErrors OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of ICMP messages which the entity received but
            determined as having ICMP-specific errors (bad ICMP
            checksums, bad length, etc.)."
    ::= { icmp 2 }

icmpInDestUnreachs OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of ICMP Destination Unreachable messages
            received."
    ::= { icmp 3 }

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

```
    STATUS     deprecated
    DESCRIPTION
           "The number of ICMP Time Exceeded messages received."
    ::= { icmp 4 }

icmpInParmProbs OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of ICMP Parameter Problem messages received."
    ::= { icmp 5 }

icmpInSrcQuenchs OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of ICMP Source Quench messages received."
    ::= { icmp 6 }

icmpInRedirects OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of ICMP Redirect messages received."
    ::= { icmp 7 }

icmpInEchos OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of ICMP Echo (request) messages received."
    ::= { icmp 8 }

icmpInEchoReps OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of ICMP Echo Reply messages received."
    ::= { icmp 9 }

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

```
    STATUS     deprecated
    DESCRIPTION
           "The number of ICMP Timestamp (request) messages received."
    ::= { icmp 10 }

icmpInTimestampReps OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of ICMP Timestamp Reply messages received."
    ::= { icmp 11 }

icmpInAddrMasks OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of ICMP Address Mask Request messages received."
    ::= { icmp 12 }

icmpInAddrMaskReps OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of ICMP Address Mask Reply messages received."
    ::= { icmp 13 }

icmpOutMsgs OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The total number of ICMP messages which this entity
            attempted to send.  Note that this counter includes all
            those counted by icmpOutErrors."
    ::= { icmp 14 }

icmpOutErrors OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of ICMP messages which this entity did not send
            due to problems discovered within ICMP such as a lack of
            buffers.  This value should not include errors discovered
            outside the ICMP layer such as the inability of IP to route
```

```
            the resultant datagram.  In some implementations there may
            be no types of error which contribute to this counter's
            value."
    ::= { icmp 15 }

icmpOutDestUnreachs OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of ICMP Destination Unreachable messages sent."
    ::= { icmp 16 }

icmpOutTimeExcds OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of ICMP Time Exceeded messages sent."
    ::= { icmp 17 }

icmpOutParmProbs OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of ICMP Parameter Problem messages sent."
    ::= { icmp 18 }

icmpOutSrcQuenchs OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of ICMP Source Quench messages sent."
    ::= { icmp 19 }

icmpOutRedirects OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of ICMP Redirect messages sent.  For a host, this
            object will always be zero, since hosts do not send
            redirects."
    ::= { icmp 20 }

icmpOutEchos OBJECT-TYPE
```

```
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of ICMP Echo (request) messages sent."
    ::= { icmp 21 }

icmpOutEchoReps OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of ICMP Echo Reply messages sent."
    ::= { icmp 22 }

icmpOutTimestamps OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of ICMP Timestamp (request) messages sent."
    ::= { icmp 23 }

icmpOutTimestampReps OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of ICMP Timestamp Reply messages sent."
    ::= { icmp 24 }

icmpOutAddrMasks OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of ICMP Address Mask Request messages sent."
    ::= { icmp 25 }

icmpOutAddrMaskReps OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     deprecated
    DESCRIPTION
           "The number of ICMP Address Mask Reply messages sent."
    ::= { icmp 26 }

-- conformance information
```

```
ipMIBConformance OBJECT IDENTIFIER ::= { ipMIB 2 }

ipMIBCompliances OBJECT IDENTIFIER ::= { ipMIBConformance 1 }
ipMIBGroups      OBJECT IDENTIFIER ::= { ipMIBConformance 2 }

ipv6Conformance OBJECT IDENTIFIER ::= { ipv6MIB 3 }

ipv6Compliances OBJECT IDENTIFIER ::= { ipv6Conformance 1 }
ipv6Groups      OBJECT IDENTIFIER ::= { ipv6Conformance 2 }


-- ipv6IcmpConformance OBJECT IDENTIFIER ::= { ipv6IcmpMIB 2 }
--
-- ipv6IcmpCompliances OBJECT IDENTIFIER ::= { ipv6IcmpConformance 1 }
-- ipv6IcmpGroups      OBJECT IDENTIFIER ::= { ipv6IcmpConformance 2 }


-- compliance statements

ipMIBCompliance MODULE-COMPLIANCE
    STATUS     deprecated
    DESCRIPTION
           "The compliance statement for systems which implement only
            IPv4.  For version-independence, this compliance statement
            is deprecated in favor of ipMIBCompliance2."
    MODULE  -- this module
        MANDATORY-GROUPS { ipGroup,
                           icmpGroup }
    ::= { ipMIBCompliances 1 }

-- ipv6Compliance MODULE-COMPLIANCE
-- .ST c
-- .(D
-- "The compliance statement for systems which
-- implement ipv6 MIB."
-- .)D
--             MODULE  -- -- this module
--                 MANDATORY-GROUPS { ipv6GeneralGroup,
--                                    ipv6NotificationGroup }
--                   OBJECT    ipv6Forwarding
--                     MIN-ACCESS  read-only
-- .(D
-- "An agent is not required to provide write
-- access to this object"
-- .)D
--                   OBJECT    ipv6DefaultHopLimit
--                     MIN-ACCESS  read-only
-- .(D
```

```
-- "An agent is not required to provide write
-- access to this object"
-- .)D
--                   OBJECT    ipv6IfDescr
--                     MIN-ACCESS  read-only
-- .(D
-- "An agent is not required to provide write
-- access to this object"
-- .)D
--                   OBJECT    ipv6IfIdentifier
--                     MIN-ACCESS  read-only
-- .(D
-- "An agent is not required to provide write
-- access to this object"
-- .)D
--                   OBJECT    ipv6IfIdentifierLength
--                     MIN-ACCESS  read-only
-- .(D
-- "An agent is not required to provide write
-- access to this object"
-- .)D
--
--                   OBJECT    ipv6IfAdminStatus
--                     MIN-ACCESS  read-only
-- .(D
-- "An agent is not required to provide write
-- access to this object"
-- .)D
--                   OBJECT    ipv6RouteValid
--                     MIN-ACCESS  read-only
-- .(D
-- "An agent is not required to provide write
-- access to this object"
-- .)D
--                   OBJECT    ipv6NetToMediaValid
--                     MIN-ACCESS  read-only
-- .(D
-- "An agent is not required to provide write
-- access to this object"
-- .)D
--             ::= { ipv6Compliances 1 }



-- units of conformance

ipGroup2 OBJECT-GROUP
    OBJECTS   { ipForwarding, ipDefaultTTL }
```

```
    STATUS     current
    DESCRIPTION
           "The group of IPv4-specific objects for basic management of
            IP entities."
    ::= { ipMIBGroups 3 }

-- I'm defining too many groups.

ipIfStatsGroup OBJECT-GROUP
    OBJECTS   { ipIfStatsInReceives, ipIfStatsInHdrErrors,
                                ipIfStatsInTooBigErrors, ipIfStatsInNoRoutes,
                                ipIfStatsInAddrErrors,
ipIfStatsInUnknownProtos,
                                ipIfStatsInTruncatedPkts, ipIfStatsInDiscards,
                                ipIfStatsInDelivers,
ipIfStatsOutForwDatagrams,
                                ipIfStatsOutRequests, ipIfStatsOutDiscards,
                                ipIfStatsOutFragOKs, ipIfStatsOutFragFails,
                                ipIfStatsOutFragCreates, ipIfStatsReasmReqds,
                                ipIfStatsReasmOKs, ipIfStatsReasmFails,
                                ipIfStatsInMcastPkts, ipIfStatsOutMcastPkts }
    STATUS     current
    DESCRIPTION
           "IP per-interface or per-system statistics."
    ::= { ipMIBGroups 4 }

-- XXX some HC statistics groups

ipv6ScopeGroup OBJECT-GROUP
    OBJECTS   { ipv6ScopeIdLinkLocal, ipv6ScopeIdSubnetLocal,
                ipv6ScopeIdAdminLocal, ipv6ScopeIdSiteLocal,
                                ipv6ScopeId6, ipv6ScopeId7,
                                ipv6ScopeIdOrganizationLocal, ipv6ScopeId9,
                                ipv6ScopeIdA, ipv6ScopeIdB,
                                ipv6ScopeIdC, ipv6ScopeIdD }
    STATUS     current
    DESCRIPTION
           "The group of objects for managing IPv6 scope zones."
    ::= { ipMIBGroups 5 }

ipGroup OBJECT-GROUP
    OBJECTS   { ipForwarding, ipDefaultTTL, ipInReceives,
                ipInHdrErrors, ipInAddrErrors,
                ipForwDatagrams, ipInUnknownProtos,
                ipInDiscards, ipInDelivers, ipOutRequests,
                ipOutDiscards, ipOutNoRoutes,
                ipReasmTimeout, ipReasmReqds, ipReasmOKs,
                ipReasmFails, ipFragOKs,
```

```
                ipFragFails, ipFragCreates,
                ipAdEntAddr, ipAdEntIfIndex, ipAdEntNetMask,
```

```
                ipAdEntBcastAddr, ipAdEntReasmMaxSize,
                ipNetToMediaIfIndex, ipNetToMediaPhysAddress,
                ipNetToMediaNetAddress, ipNetToMediaType,
                ipRoutingDiscards }
    STATUS     deprecated
    DESCRIPTION
           "The ip group of objects providing for basic management of IP
            entities, exclusive of the management of IP routes."
    ::= { ipMIBGroups 1 }

icmpGroup OBJECT-GROUP
    OBJECTS   { icmpInMsgs, icmpInErrors,
                icmpInDestUnreachs, icmpInTimeExcds,
                icmpInParmProbs, icmpInSrcQuenchs,
                icmpInRedirects, icmpInEchos,
                icmpInEchoReps, icmpInTimestamps,
                icmpInTimestampReps, icmpInAddrMasks,
                icmpInAddrMaskReps, icmpOutMsgs,
                icmpOutErrors, icmpOutDestUnreachs,
                icmpOutTimeExcds, icmpOutParmProbs,
                icmpOutSrcQuenchs, icmpOutRedirects,
                icmpOutEchos, icmpOutEchoReps,
                icmpOutTimestamps, icmpOutTimestampReps,
                icmpOutAddrMasks, icmpOutAddrMaskReps }
    STATUS     deprecated
    DESCRIPTION
           "The icmp group of objects providing ICMP statistics."
    ::= { ipMIBGroups 2 }

ipv6GeneralGroup2 OBJECT-GROUP
    OBJECTS { ipv6Forwarding,
              ipv6DefaultHopLimit }
    STATUS     current
    DESCRIPTION
           "The IPv6 group of objects providing for basic management of
            IPv6 entities."
    ::= { ipv6Groups 3 }


END
```

5. Open Issues / To Do

Any other objects from ipv6IfTable that we need? What's ipv6IfEffectiveMtu good for? When can ipv6IfPhysicalAddress be different than the interface's ifPhysAddress? ipv6IfOperStatus?

the ipv6IfTable could have one ipv6InterfaceIndex per address to allow keeping stats per address. Should we allow for this?

Verbatim from meeting notes -- these notes were too concise for me to remember what they meant:
IPv6: ipv6AddrPrefixTable is v6-specific
v4: subnets on interface
so redo this table
add origin type
maybe one table which lists addresses
and a stateless autoconf table which sparsely augments the prefix table

ipIfStatsTable: Add octet counters similar to ifTable, ifXTable and ipMRouteInterfaceTable? e.g. inOctets outOctets inBcastPkts outBcastPkts HCInOctets HCInUcastPkts HCInMcastPkts HCInBcastPkts HCOutOctets HCOutUcastPkts HCOutMcastPkts HCOutBcastPkts InMcastOctets OutMcastOctets HCInMcastOctets HCOutMcastOctets

inetNetToMediaState: what values for !ipv6? Why no value for incomplete?

How to describe what stats are required, especially in the ICMP Msg table? Require per-interface, per-system, both, some? Require tracking every ICMP message type or just the ones the system cares about? What about ICMP codes?

Should the ipv6 scope table have a scope name string like ipMRouteScopeNameString?

Need to update conformance info.

Note: more open issues / to do items scattered in comments in MIB.

6. Acknoledgments

This document contains objects modified from RFC 1213 [1], RFC 2011 [2], RFC 2465 [4], and RFC 2466 [5].

7. References

[1] Rose, M. and K. McCloghrie, "Management Information Base for Network Management of TCP/IP-based internets", RFC 1213, March 1991.

[2] K. McCloghrie, "SNMPv2 Management Information Base for the Internet Protocol using SMIv2", RFC 2011, November 1996.

[3] Narten, T., E. Nordmark and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December 1998.

[4] Haskin, D. and S. Onishi, "Management Information Base for IP Version 6: Textual Conventions and General Group", RFC 2465, December 1998.

[5] Haskin, D. and S. Onishi, "Management Information Base for IP Version 6: ICMPv6 Group", RFC 2466, December 1998.

[6] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", RFC 2571, April 1999.

[7] Rose, M., and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", STD 16, RFC 1155, May 1990.

[8] Rose, M., and K. McCloghrie, "Concise MIB Definitions", STD 16, RFC 1212, March 1991.

[9] Rose, M., "A Convention for Defining Traps for use with the SNMP", RFC 1215, March 1991.

[10] 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.

[11] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M., and S. Waldbusser, "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[12] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M., and S. Waldbusser, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

[13] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol", STD 15, RFC 1157, May 1990.

[14] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Introduction to Community-based SNMPv2", RFC 1901, January 1996.

[15] 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.

[16] Case, J., Harrington D., Presuhn R., and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", RFC 2572, April 1999.

[17] Blumenthal, U., and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", RFC 2574, April 1999.

[18] 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.

[19] Levi, D., Meyer, P., and B. Stewart, "SNMPv3 Applications", RFC 2573, April 1999.

[20] Wijnen, B., Presuhn, R., and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", RFC 2575, April 1999.

[21] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction to Version 3 of the Internet-standard Network Management Framework", RFC 2570, April 1999.

8. 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.

There are a number of managed objects in this MIB that may contain sensitive information. These are:

-- not yet evaluated

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 [17] and the View-based Access Control Model RFC 2575 [20] 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.

9. Editor's Address

Bill Fenner
AT&T Labs -- Research
75 Willow Rd
Menlo Park, CA 94025
USA

Email: fenner@research.att.com

10. Full Copyright Statement