

SNMPv2 Management Information Base
for the Internet Protocol using SMIPv2

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

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.

Table of Contents

1. Introduction	1
2. Definitions	2
2.1 The IP Group	3
2.2 The ICMP Group.....	11
2.3 Conformance Information	16
2.3.1 Compliance Statements	16
2.3.2 Units of Conformance	16
3. Acknowledgements	18
4. References	18
5. Security Considerations	18
6. Editor's Address	18

1. Introduction

A management system contains: several (potentially many) nodes, each with a processing entity, termed an agent, which has access to management instrumentation; at least one management station; and, a management protocol, used to convey management information between the agents and management stations. Operations of the protocol are carried out under an administrative framework which defines authentication, authorization, access control, and privacy policies.

Management stations execute management applications which monitor and control managed elements. Managed elements are devices such as hosts, routers, terminal servers, etc., which are monitored and controlled via access to their management information.

Management information is viewed as a collection of managed objects, residing in a virtual information store, termed the Management Information Base (MIB). Collections of related objects are defined in MIB modules. These modules are written using a subset of OSI's Abstract Syntax Notation One (ASN.1) [1], termed the Structure of Management Information (SMI) [2].

This document is the MIB module which defines managed objects for managing implementations of the Internet Protocol (IP) [3] and its associated Internet Control Message Protocol (ICMP) [4].

The managed objects in this MIB module were originally defined using the SNMPv1 framework as a part of MIB-II [5]. Since then, the managed objects related to managing routes in an IP internet were updated by RFC 1354 [6]. This document takes the remaining MIB-II objects for these protocols, and defines them using the SNMPv2 framework.

2. Definitions

```
IP-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
    MODULE-IDENTITY, OBJECT-TYPE, Integer32,  
    Counter32, IpAddress, mib-2          FROM SNMPv2-SMI  
    PhysAddress                          FROM SNMPv2-TC  
    MODULE-COMPLIANCE, OBJECT-GROUP     FROM SNMPv2-CONF;
```

```
ipMIB MODULE-IDENTITY
```

```
    LAST-UPDATED "9411010000Z"  
    ORGANIZATION "IETF SNMPv2 Working Group"  
    CONTACT-INFO  
        "          Keith McCloghrie
```

```
        Postal: Cisco Systems, Inc.  
                170 West Tasman Drive  
                San Jose, CA 95134-1706  
                US
```

```
        Phone:  +1 408 526 5260  
        Email:  kzm@cisco.com"
```

```
DESCRIPTION
    "The MIB module for managing IP and ICMP implementations,
    but excluding their management of IP routes."
REVISION      "9103310000Z"
DESCRIPTION
    "The initial revision of this MIB module was part of MIB-
    II."
 ::= { mib-2 48 }

-- the IP 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 IP
        router in respect to the forwarding of datagrams received
        by, but not addressed to, this entity.  IP routers forward
        datagrams.  IP 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 IP header of datagrams originated at this entity,
        whenever a TTL value is not supplied by the transport layer
        protocol."
```

```
 ::= { ip 2 }

ipInReceives OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    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 current

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 current

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 current

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 current

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 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."

```
::= { ip 8 }
```

ipInDelivers OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

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 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 ipForwDatagrams."

```
::= { ip 10 }
```

ipOutDiscards 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 ipForwDatagrams if any such packets met this (discretionary) discard criterion."

```
::= { ip 11 }
```

ipOutNoRoutes OBJECT-TYPE

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

"The number of IP datagrams successfully re-assembled."

::= { ip 15 }

ipReasmFails 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."  
 ::= { ip 16 }
```

```
ipFragOKs OBJECT-TYPE
```

```
SYNTAX Counter32
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
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 current
```

```
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 current
```

```
DESCRIPTION
```

```
    "The number of IP datagram fragments that have been  
    generated as a result of fragmentation at this entity."
```

```
 ::= { ip 19 }
```

```
-- the IP address table
```

```
ipAddrTable OBJECT-TYPE
```

```
SYNTAX SEQUENCE OF IpAddrEntry
```

```
MAX-ACCESS not-accessible
```

```
STATUS current
```

```
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 current
```

```
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      current
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      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 1573's ifIndex."
 ::= { ipAddrEntry 2 }

ipAdEntNetMask OBJECT-TYPE
SYNTAX      IpAddress
MAX-ACCESS  read-only
STATUS      current
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      current
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      current
```

```
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 }
```

```
-- ipRouteTable ::= { ip 21 }      obsolete
```

```
-- the 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      current
```

```
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      current
```

```
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      current
```

```
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 1573's ifIndex."
```

```
 ::= { ipNetToMediaEntry 1 }
```

```
ipNetToMediaPhysAddress OBJECT-TYPE
```

```
SYNTAX      PhysAddress
```

```
MAX-ACCESS  read-create
```

```
STATUS      current
```

```
DESCRIPTION
```

```
    "The media-dependent 'physical' address."
```

```
 ::= { ipNetToMediaEntry 2 }
```

```
ipNetToMediaNetAddress OBJECT-TYPE
```

```
SYNTAX      IpAddress
```

```
MAX-ACCESS  read-create
```

```
STATUS      current
```

```
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      current
```

```
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 }
```

```
ipRoutingDiscards OBJECT-TYPE
```

```
SYNTAX Counter32
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
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 ICMP group
```

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

```
icmpInMsgs 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 icmpInErrors."

```
::= { icmp 1 }
```

```
icmpInErrors 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.)."

```
::= { icmp 2 }
```

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

icmpInTimeExcds OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of ICMP Time Exceeded messages received."
    ::= { icmp 4 }

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

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

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

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

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

icmpInTimestamps OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of ICMP Timestamp (request) messages received."
    ::= { icmp 10 }

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

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

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

icmpOutMsgs OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    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 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."
```

```
::= { icmp 15 }
```

```
icmpOutDestUnreachs OBJECT-TYPE
```

```
SYNTAX Counter32
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"The number of ICMP Destination Unreachable messages sent."
```

```
::= { icmp 16 }
```

```
icmpOutTimeExcds OBJECT-TYPE
```

```
SYNTAX Counter32
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"The number of ICMP Time Exceeded messages sent."
```

```
::= { icmp 17 }
```

```
icmpOutParmProbs OBJECT-TYPE
```

```
SYNTAX Counter32
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"The number of ICMP Parameter Problem messages sent."
```

```
::= { icmp 18 }
```

```
icmpOutSrcQuenchs OBJECT-TYPE
```

```
SYNTAX Counter32
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"The number of ICMP Source Quench messages sent."
```

```
::= { icmp 19 }
```

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

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

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

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

icmpOutAddrMasks OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of ICMP Address Mask Request messages sent."
```

```

 ::= { icmp 25 }

icmpOutAddrMaskReps OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    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 }

-- compliance statements

ipMIBCompliance MODULE-COMPLIANCE
    STATUS      current
    DESCRIPTION
        "The compliance statement for SNMPv2 entities which
         implement IP."
    MODULE      -- this module
        MANDATORY-GROUPS { ipGroup,
                           icmpGroup }
 ::= { ipMIBCompliances 1 }

-- units of conformance

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      current
    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 current
```

```
  DESCRIPTION
```

```
    "The icmp group of objects providing ICMP statistics."
 ::= { ipMIBGroups 2 }
```

```
END
```

3. Acknowledgements

This document contains a modified subset of RFC 1213.

4. References

- [1] Information processing systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1), International Organization for Standardization. International Standard 8824, (December, 1987).
- [2] McCloghrie, K., Editor, "Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1902, Cisco Systems, January 1996.
- [3] Postel, J., "Internet Protocol - DARPA Internet Program Protocol Specification", STD 5, RFC 791, DARPA, September 1981.
- [4] Postel, J., "Internet Control Message Protocol - DARPA Internet Program Protocol Specification", STD 5, RFC 792, USC/Information Sciences Institute, September 1981.
- [5] McCloghrie, K., and M. Rose, "Management Information Base for Network Management of TCP/IP-based internets: MIB-II", STD 17, RFC 1213, March 1991.
- [6] Baker, F., "IP Forwarding Table MIB", RFC 1354, ACC, July 1992.

5. Security Considerations

Security issues are not discussed in this memo.

6. Editor's Address

Keith McCloghrie
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
US

Phone: +1 408 526 5260
EMail: kzm@cisco.com

