

Network Working Group
Request for Comments: 2932
Category: Standards Track

K. McCloghrie
cisco Systems
D. Farinacci
Procket Networks
D. Thaler
Microsoft
October 2000

IPv4 Multicast Routing MIB

Status of this Memo

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

Copyright Notice

Copyright (C) The Internet Society (2000). All Rights Reserved.

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 managing IP Multicast Routing for IPv4, independent of the specific multicast routing protocol in use.

Table of Contents

1	Introduction	2
2	The SNMP Management Framework	2
3	Overview	3
4	Definitions	4
5	IANA Considerations	22
6	Security Considerations	22
7	Intellectual Property Notice	23
8	Acknowledgements	23
9	Authors' Addresses	24
10	References	25
11	Full Copyright Statement	27

1. Introduction

This MIB describes objects used for managing IP Multicast Routing [16], independent of the specific multicast routing protocol [17-21] in use. Managed objects specific to particular multicast routing protocols are specified elsewhere. Similarly, this MIB does not support management of multicast routing for other address families, including IPv6. Such management may be supported by other MIBs.

2. The SNMP Management Framework

The SNMP Management Framework presently consists of five major components:

- o An overall architecture, described in RFC 2571 [1].
- 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 [2], STD 16, RFC 1212 [3] and RFC 1215 [4]. The second version, called SMIV2, is described in STD 58, RFC 2578 [5], STD 58, RFC 2579 [6] and STD 58, RFC 2580 [7].
- 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 [8]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901 [9] and RFC 1906 [10]. The third version of the message protocol is called SNMPv3 and described in RFC 1906 [10], RFC 2572 [11] and RFC 2574 [12].
- o Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in STD 15, RFC 1157 [8]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [13].
- o A set of fundamental applications described in RFC 2573 [14] and the view-based access control mechanism described in RFC 2575 [15].

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.

3. Overview

This MIB module contains one scalar and five tables. The tables are:

- (1) the IP Multicast Route Table containing multicast routing information for IP datagrams sent by particular sources to the IP multicast groups known to a router.
- (2) the IP Multicast Routing Next Hop Table containing information on the next-hops for the routing IP multicast datagrams. Each entry is one of a list of next-hops on outgoing interfaces for particular sources sending to a particular multicast group address.
- (3) the IP Multicast Routing Interface Table containing multicast routing information specific to interfaces.
- (4) the IP Multicast Scope Boundary Table containing the boundaries configured for multicast scopes [22].
- (5) the IP Multicast Scope Name Table containing human-readable names of multicast scope.

4. Definitions

```
IPMROUTE-STD-MIB DEFINITIONS ::= BEGIN
```

IMPORTS

```

MODULE-IDENTITY, OBJECT-TYPE, mib-2,
Integer32, Counter32, Counter64, Gauge32,
IpAddress, TimeTicks          FROM SNMPv2-SMI
RowStatus, TEXTUAL-CONVENTION,
TruthValue                    FROM SNMPv2-TC
MODULE-COMPLIANCE, OBJECT-GROUP FROM SNMPv2-CONF
SnmAdminString                FROM SNMP-FRAMEWORK-MIB
InterfaceIndexOrZero,
InterfaceIndex                FROM IF-MIB
IANAipRouteProtocol,
IANAipMRouteProtocol          FROM IANA-RTPROTO-MIB;
```

ipMRouteStdMIB MODULE-IDENTITY

```
LAST-UPDATED "200009220000Z" -- September 22, 2000
```

```
ORGANIZATION "IETF IDMR Working Group"
```

CONTACT-INFO

```

" Dave Thaler
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399
US
```

```
Phone: +1 425 703 8835
```

```
E-Mail: dthaler@microsoft.com"
```

DESCRIPTION

```
"The MIB module for management of IP Multicast routing, but
independent of the specific multicast routing protocol in
use."
```

```
REVISION "200009220000Z" -- September 22, 2000
```

DESCRIPTION

```
"Initial version, published as RFC 2932."
```

```
::= { mib-2 83 }
```

-- Textual Conventions

```
LanguageTag ::= TEXTUAL-CONVENTION
```

```
DISPLAY-HINT "100a"
```

```
STATUS current
```

DESCRIPTION

```
"An RFC 1766-style language tag, with all alphabetic
characters converted to lowercase. This restriction is
intended to make the lexical ordering imposed by SNMP useful
```

when applied to language tags. Note that it is theoretically possible for a valid language tag to exceed the allowed length of this syntax, and thus be impossible to represent with this syntax. Sampling of language tags in current use on the Internet suggests that this limit does not pose a serious problem in practice."

SYNTAX OCTET STRING (SIZE (1..100))

-- Top-level structure of the MIB

ipMRouteMIBObjects OBJECT IDENTIFIER ::= { ipMRouteStdMIB 1 }

ipMRoute OBJECT IDENTIFIER ::= { ipMRouteMIBObjects 1 }

-- the IP Multicast Routing MIB-Group

--

-- a collection of objects providing information about

-- IP Multicast Groups

ipMRouteEnable OBJECT-TYPE

SYNTAX INTEGER { enabled(1), disabled(2) }

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The enabled status of IP Multicast routing on this router."

::= { ipMRoute 1 }

ipMRouteEntryCount OBJECT-TYPE

SYNTAX Gauge32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of rows in the ipMRouteTable. This can be used to monitor the multicast routing table size."

::= { ipMRoute 7 }

ipMRouteTable OBJECT-TYPE

SYNTAX SEQUENCE OF IpMRouteEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The (conceptual) table containing multicast routing information for IP datagrams sent by particular sources to the IP multicast groups known to this router."

::= { ipMRoute 2 }

```

ipMRouteEntry OBJECT-TYPE
    SYNTAX      IpMRouteEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry (conceptual row) containing the multicast routing
        information for IP datagrams from a particular source and
        addressed to a particular IP multicast group address.
        Discontinuities in counters in this entry can be detected by
        observing the value of ipMRouteUpTime."
    INDEX       { ipMRouteGroup,
                  ipMRouteSource,
                  ipMRouteSourceMask }
    ::= { ipMRouteTable 1 }

```

```

IpMRouteEntry ::= SEQUENCE {
    ipMRouteGroup          IpAddress,
    ipMRouteSource         IpAddress,
    ipMRouteSourceMask     IpAddress,
    ipMRouteUpstreamNeighbor IpAddress,
    ipMRouteInIfIndex      InterfaceIndexOrZero,
    ipMRouteUpTime         TimeTicks,
    ipMRouteExpiryTime     TimeTicks,
    ipMRoutePkts           Counter32,
    ipMRouteDifferentInIfPackets Counter32,
    ipMRouteOctets         Counter32,
    ipMRouteProtocol       IANAipMRouteProtocol,
    ipMRouteRtProto        IANAipRouteProtocol,
    ipMRouteRtAddress      IpAddress,
    ipMRouteRtMask         IpAddress,
    ipMRouteRtType         INTEGER,
    ipMRouteHCOctets       Counter64
}

```

```

ipMRouteGroup OBJECT-TYPE
    SYNTAX      IpAddress
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The IP multicast group address for which this entry
        contains multicast routing information."
    ::= { ipMRouteEntry 1 }

```

```

ipMRouteSource OBJECT-TYPE
    SYNTAX      IpAddress
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION

```

"The network address which when combined with the corresponding value of ipMRouteSourceMask identifies the sources for which this entry contains multicast routing information."

::= { ipMRouteEntry 2 }

ipMRouteSourceMask OBJECT-TYPE

SYNTAX IpAddress
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION

"The network mask which when combined with the corresponding value of ipMRouteSource identifies the sources for which this entry contains multicast routing information."

::= { ipMRouteEntry 3 }

ipMRouteUpstreamNeighbor OBJECT-TYPE

SYNTAX IpAddress
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"The address of the upstream neighbor (e.g., RPF neighbor) from which IP datagrams from these sources to this multicast address are received, or 0.0.0.0 if the upstream neighbor is unknown (e.g., in CBT)."

::= { ipMRouteEntry 4 }

ipMRouteInIfIndex OBJECT-TYPE

SYNTAX InterfaceIndexOrZero
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"The value of ifIndex for the interface on which IP datagrams sent by these sources to this multicast address are received. A value of 0 indicates that datagrams are not subject to an incoming interface check, but may be accepted on multiple interfaces (e.g., in CBT)."

::= { ipMRouteEntry 5 }

ipMRouteUpTime OBJECT-TYPE

SYNTAX TimeTicks
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"The time since the multicast routing information represented by this entry was learned by the router."

::= { ipMRouteEntry 6 }

```
ipMRouteExpiryTime OBJECT-TYPE
    SYNTAX      TimeTicks
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The minimum amount of time remaining before this entry will
        be aged out.  The value 0 indicates that the entry is not
        subject to aging."
    ::= { ipMRouteEntry 7 }

ipMRoutePkts OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of packets which this router has received from
        these sources and addressed to this multicast group
        address."
    ::= { ipMRouteEntry 8 }

ipMRouteDifferentInIfPackets OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of packets which this router has received from
        these sources and addressed to this multicast group address,
        which were dropped because they were not received on the
        interface indicated by ipMRouteInIfIndex.  Packets which are
        not subject to an incoming interface check (e.g., using CBT)
        are not counted."
    ::= { ipMRouteEntry 9 }

ipMRouteOctets OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of octets contained in IP datagrams which were
        received from these sources and addressed to this multicast
        group address, and which were forwarded by this router."
    ::= { ipMRouteEntry 10 }

ipMRouteProtocol OBJECT-TYPE
    SYNTAX      IANAipMRouteProtocol
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
```

"The multicast routing protocol via which this multicast forwarding entry was learned."
 ::= { ipMRouteEntry 11 }

ipMRouterRtProto OBJECT-TYPE

SYNTAX IANAipRouteProtocol

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The routing mechanism via which the route used to find the upstream or parent interface for this multicast forwarding entry was learned. Inclusion of values for routing protocols is not intended to imply that those protocols need be supported."

::= { ipMRouteEntry 12 }

ipMRouterRtAddress OBJECT-TYPE

SYNTAX IPAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The address portion of the route used to find the upstream or parent interface for this multicast forwarding entry."

::= { ipMRouteEntry 13 }

ipMRouterRtMask OBJECT-TYPE

SYNTAX IPAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The mask associated with the route used to find the upstream or parent interface for this multicast forwarding entry."

::= { ipMRouteEntry 14 }

ipMRouterRtType OBJECT-TYPE

SYNTAX INTEGER {
 unicast (1), -- Unicast route used in multicast RIB
 multicast (2) -- Multicast route
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The reason the given route was placed in the (logical) multicast Routing Information Base (RIB). A value of unicast means that the route would normally be placed only in the unicast RIB, but was placed in the multicast RIB (instead or in addition) due to local configuration, such as when running PIM over RIP. A value of multicast means that

the route was explicitly added to the multicast RIB by the routing protocol, such as DVMRP or Multiprotocol BGP."

```
::= { ipMRouteEntry 15 }
```

```
ipMRouteHCOctets OBJECT-TYPE
```

```
SYNTAX Counter64
```

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

```
STATUS current
```

```
DESCRIPTION
```

"The number of octets contained in IP datagrams which were received from these sources and addressed to this multicast group address, and which were forwarded by this router.

This object is a 64-bit version of ipMRouteOctets."

```
::= { ipMRouteEntry 16 }
```

```
--
```

```
-- The IP Multicast Routing Next Hop Table
```

```
--
```

```
ipMRouteNextHopTable OBJECT-TYPE
```

```
SYNTAX SEQUENCE OF IpMRouteNextHopEntry
```

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

```
STATUS current
```

```
DESCRIPTION
```

"The (conceptual) table containing information on the next-hops on outgoing interfaces for routing IP multicast datagrams. Each entry is one of a list of next-hops on outgoing interfaces for particular sources sending to a particular multicast group address."

```
::= { ipMRoute 3 }
```

```
ipMRouteNextHopEntry OBJECT-TYPE
```

```
SYNTAX IpMRouteNextHopEntry
```

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

```
STATUS current
```

```
DESCRIPTION
```

"An entry (conceptual row) in the list of next-hops on outgoing interfaces to which IP multicast datagrams from particular sources to a IP multicast group address are routed. Discontinuities in counters in this entry can be detected by observing the value of ipMRouteUpTime."

```
INDEX { ipMRouteNextHopGroup, ipMRouteNextHopSource,
        ipMRouteNextHopSourceMask, ipMRouteNextHopIfIndex,
        ipMRouteNextHopAddress }
```

```
::= { ipMRouteNextHopTable 1 }
```

```
IpMRouteNextHopEntry ::= SEQUENCE {
```

```
    ipMRouteNextHopGroup          IpAddress,
```

```

ipMRouteNextHopSource          IPAddress,
ipMRouteNextHopSourceMask     IPAddress,
ipMRouteNextHopIfIndex       InterfaceIndex,
ipMRouteNextHopAddress        IPAddress,
ipMRouteNextHopState          INTEGER,
ipMRouteNextHopUpTime         TimeTicks,
ipMRouteNextHopExpiryTime     TimeTicks,
ipMRouteNextHopClosestMemberHops Integer32,
ipMRouteNextHopProtocol       IANAipMRouteProtocol,
ipMRouteNextHopPkts           Counter32
}

```

ipMRouteNextHopGroup OBJECT-TYPE

SYNTAX IPAddress

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The IP multicast group for which this entry specifies a next-hop on an outgoing interface."

::= { ipMRouteNextHopEntry 1 }

ipMRouteNextHopSource OBJECT-TYPE

SYNTAX IPAddress

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The network address which when combined with the corresponding value of ipMRouteNextHopSourceMask identifies the sources for which this entry specifies a next-hop on an outgoing interface."

::= { ipMRouteNextHopEntry 2 }

ipMRouteNextHopSourceMask OBJECT-TYPE

SYNTAX IPAddress

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The network mask which when combined with the corresponding value of ipMRouteNextHopSource identifies the sources for which this entry specifies a next-hop on an outgoing interface."

::= { ipMRouteNextHopEntry 3 }

ipMRouteNextHopIfIndex OBJECT-TYPE

SYNTAX InterfaceIndex

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The ifIndex value of the interface for the outgoing interface for this next-hop."

::= { ipMRouteNextHopEntry 4 }

ipMRouteNextHopAddress OBJECT-TYPE

SYNTAX IPAddress

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The address of the next-hop specific to this entry. For most interfaces, this is identical to ipMRouteNextHopGroup. NBMA interfaces, however, may have multiple next-hop addresses out a single outgoing interface."

::= { ipMRouteNextHopEntry 5 }

ipMRouteNextHopState OBJECT-TYPE

SYNTAX INTEGER { pruned(1), forwarding(2) }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"An indication of whether the outgoing interface and next-hop represented by this entry is currently being used to forward IP datagrams. The value 'forwarding' indicates it is currently being used; the value 'pruned' indicates it is not."

::= { ipMRouteNextHopEntry 6 }

ipMRouteNextHopUpTime OBJECT-TYPE

SYNTAX TimeTicks

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The time since the multicast routing information represented by this entry was learned by the router."

::= { ipMRouteNextHopEntry 7 }

ipMRouteNextHopExpiryTime OBJECT-TYPE

SYNTAX TimeTicks

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The minimum amount of time remaining before this entry will be aged out. If ipMRouteNextHopState is pruned(1), the remaining time until the prune expires and the state reverts to forwarding(2). Otherwise, the remaining time until this entry is removed from the table. The time remaining may be copied from ipMRouteExpiryTime if the protocol in use for this entry does not specify next-hop timers. The value 0

indicates that the entry is not subject to aging."
 ::= { ipMRouteNextHopEntry 8 }

ipMRouteNextHopClosestMemberHops OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The minimum number of hops between this router and any member of this IP multicast group reached via this next-hop on this outgoing interface. Any IP multicast datagrams for the group which have a TTL less than this number of hops will not be forwarded to this next-hop."

::= { ipMRouteNextHopEntry 9 }

ipMRouteNextHopProtocol OBJECT-TYPE

SYNTAX IANAipMRouteProtocol

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The routing mechanism via which this next-hop was learned."

::= { ipMRouteNextHopEntry 10 }

ipMRouteNextHopPkts OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of packets which have been forwarded using this route."

::= { ipMRouteNextHopEntry 11 }

--

-- The Multicast Routing Interface Table

--

ipMRouteInterfaceTable OBJECT-TYPE

SYNTAX SEQUENCE OF IpMRouteInterfaceEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The (conceptual) table containing multicast routing information specific to interfaces."

::= { ipMRoute 4 }

ipMRouteInterfaceEntry OBJECT-TYPE

SYNTAX IpMRouteInterfaceEntry

MAX-ACCESS not-accessible

```

STATUS      current
DESCRIPTION
    "An entry (conceptual row) containing the multicast routing
    information for a particular interface."
INDEX       { ipMRouteInterfaceIfIndex }
 ::= { ipMRouteInterfaceTable 1 }

```

```

IpMRouteInterfaceEntry ::= SEQUENCE {
    ipMRouteInterfaceIfIndex      InterfaceIndex,
    ipMRouteInterfaceTtl          Integer32,
    ipMRouteInterfaceProtocol     IANAipMRouteProtocol,
    ipMRouteInterfaceRateLimit    Integer32,
    ipMRouteInterfaceInMcastOctets Counter32,
    ipMRouteInterfaceOutMcastOctets Counter32,
    ipMRouteInterfaceHCInMcastOctets Counter64,
    ipMRouteInterfaceHCOutMcastOctets Counter64
}

```

```

ipMRouteInterfaceIfIndex OBJECT-TYPE
    SYNTAX      InterfaceIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The ifIndex value of the interface for which this entry
        contains information."
    ::= { ipMRouteInterfaceEntry 1 }

```

```

ipMRouteInterfaceTtl OBJECT-TYPE
    SYNTAX      Integer32 (0..255)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The datagram TTL threshold for the interface. Any IP
        multicast datagrams with a TTL less than this threshold will
        not be forwarded out the interface. The default value of 0
        means all multicast packets are forwarded out the
        interface."
    ::= { ipMRouteInterfaceEntry 2 }

```

```

ipMRouteInterfaceProtocol OBJECT-TYPE
    SYNTAX      IANAipMRouteProtocol
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The routing protocol running on this interface."
    ::= { ipMRouteInterfaceEntry 3 }

```

```

ipMRouteInterfaceRateLimit OBJECT-TYPE

```

```
SYNTAX      Integer32
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    "The rate-limit, in kilobits per second, of forwarded
    multicast traffic on the interface. A rate-limit of 0
    indicates that no rate limiting is done."
DEFVAL      { 0 }
 ::= { ipMRouteInterfaceEntry 4 }
```

ipMRouteInterfaceInMcastOctets OBJECT-TYPE

```
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of octets of multicast packets that have arrived
    on the interface, including framing characters. This object
    is similar to ifInOctets in the Interfaces MIB, except that
    only multicast packets are counted."
 ::= { ipMRouteInterfaceEntry 5 }
```

ipMRouteInterfaceOutMcastOctets OBJECT-TYPE

```
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of octets of multicast packets that have been
    sent on the interface."
 ::= { ipMRouteInterfaceEntry 6 }
```

ipMRouteInterfaceHCInMcastOctets OBJECT-TYPE

```
SYNTAX      Counter64
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of octets of multicast packets that have arrived
    on the interface, including framing characters. This object
    is a 64-bit version of ipMRouteInterfaceInMcastOctets. It
    is similar to ifHCInOctets in the Interfaces MIB, except
    that only multicast packets are counted."
 ::= { ipMRouteInterfaceEntry 7 }
```

ipMRouteInterfaceHCOutMcastOctets OBJECT-TYPE

```
SYNTAX      Counter64
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of octets of multicast packets that have been
```

sent on the interface. This object is a 64-bit version of ipMRouteInterfaceOutMcastOctets."

```
::= { ipMRouteInterfaceEntry 8 }
```

```
--
```

```
-- The IP Multicast Scope Boundary Table
```

```
--
```

```
ipMRouteBoundaryTable OBJECT-TYPE
```

```
SYNTAX SEQUENCE OF IpMRouteBoundaryEntry
```

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

```
STATUS current
```

```
DESCRIPTION
```

"The (conceptual) table listing the router's scoped multicast address boundaries."

```
::= { ipMRoute 5 }
```

```
ipMRouteBoundaryEntry OBJECT-TYPE
```

```
SYNTAX IpMRouteBoundaryEntry
```

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

```
STATUS current
```

```
DESCRIPTION
```

"An entry (conceptual row) in the ipMRouteBoundaryTable representing a scoped boundary."

```
INDEX { ipMRouteBoundaryIfIndex, ipMRouteBoundaryAddress,
        ipMRouteBoundaryAddressMask }
```

```
::= { ipMRouteBoundaryTable 1 }
```

```
IpMRouteBoundaryEntry ::= SEQUENCE {
```

```
ipMRouteBoundaryIfIndex      InterfaceIndex,
```

```
ipMRouteBoundaryAddress      IpAddress,
```

```
ipMRouteBoundaryAddressMask  IpAddress,
```

```
ipMRouteBoundaryStatus       RowStatus
```

```
}
```

```
ipMRouteBoundaryIfIndex OBJECT-TYPE
```

```
SYNTAX InterfaceIndex
```

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

```
STATUS current
```

```
DESCRIPTION
```

"The IfIndex value for the interface to which this boundary applies. Packets with a destination address in the associated address/mask range will not be forwarded out this interface."

```
::= { ipMRouteBoundaryEntry 1 }
```

```
ipMRouteBoundaryAddress OBJECT-TYPE
```

```
SYNTAX IpAddress
```

```

MAX-ACCESS not-accessible
STATUS      current
DESCRIPTION
    "The group address which when combined with the
    corresponding value of ipMRouteBoundaryAddressMask
    identifies the group range for which the scoped boundary
    exists. Scoped addresses must come from the range 239.x.x.x
    as specified in RFC 2365."
 ::= { ipMRouteBoundaryEntry 2 }

```

```

ipMRouteBoundaryAddressMask OBJECT-TYPE
SYNTAX      IPAddress
MAX-ACCESS not-accessible
STATUS      current
DESCRIPTION
    "The group address mask which when combined with the
    corresponding value of ipMRouteBoundaryAddress identifies
    the group range for which the scoped boundary exists."
 ::= { ipMRouteBoundaryEntry 3 }

```

```

ipMRouteBoundaryStatus OBJECT-TYPE
SYNTAX      RowStatus
MAX-ACCESS read-create
STATUS      current
DESCRIPTION
    "The status of this row, by which new entries may be
    created, or old entries deleted from this table."
 ::= { ipMRouteBoundaryEntry 4 }

```

```

--
-- The IP Multicast Scope Name Table
--

```

```

ipMRouteScopeNameTable OBJECT-TYPE
SYNTAX      SEQUENCE OF IpMRouteScopeNameEntry
MAX-ACCESS not-accessible
STATUS      current
DESCRIPTION
    "The (conceptual) table listing the multicast scope names."
 ::= { ipMRoute 6 }

```

```

ipMRouteScopeNameEntry OBJECT-TYPE
SYNTAX      IpMRouteScopeNameEntry
MAX-ACCESS not-accessible
STATUS      current
DESCRIPTION
    "An entry (conceptual row) in the ipMRouteScopeNameTable
    representing a multicast scope name."

```

```

INDEX      { ipMRouteScopeNameAddress,
             ipMRouteScopeNameAddressMask,
             IMPLIED ipMRouteScopeNameLanguage }
 ::= { ipMRouteScopeNameTable 1 }

```

```

IpMRouteScopeNameEntry ::= SEQUENCE {
    ipMRouteScopeNameAddress      IpAddress,
    ipMRouteScopeNameAddressMask IpAddress,
    ipMRouteScopeNameLanguage    LanguageTag,
    ipMRouteScopeNameString      SnmpAdminString,
    ipMRouteScopeNameDefault     TruthValue,
    ipMRouteScopeNameStatus      RowStatus
}

```

ipMRouteScopeNameAddress OBJECT-TYPE

```

SYNTAX      IpAddress
MAX-ACCESS not-accessible
STATUS      current
DESCRIPTION
    "The group address which when combined with the
    corresponding value of ipMRouteScopeNameAddressMask
    identifies the group range associated with the multicast
    scope. Scoped addresses must come from the range
    239.x.x.x."
 ::= { ipMRouteScopeNameEntry 1 }

```

ipMRouteScopeNameAddressMask OBJECT-TYPE

```

SYNTAX      IpAddress
MAX-ACCESS not-accessible
STATUS      current
DESCRIPTION
    "The group address mask which when combined with the
    corresponding value of ipMRouteScopeNameAddress identifies
    the group range associated with the multicast scope."
 ::= { ipMRouteScopeNameEntry 2 }

```

ipMRouteScopeNameLanguage OBJECT-TYPE

```

SYNTAX      LanguageTag
MAX-ACCESS not-accessible
STATUS      current
DESCRIPTION
    "The RFC 1766-style language tag associated with the scope
    name."
 ::= { ipMRouteScopeNameEntry 3 }

```

ipMRouteScopeNameString OBJECT-TYPE

```

SYNTAX      SnmpAdminString
MAX-ACCESS read-create

```

STATUS current

DESCRIPTION

"The textual name associated with the multicast scope. The value of this object should be suitable for displaying to end-users, such as when allocating a multicast address in this scope. When no name is specified, the default value of this object should be the string 239.x.x.x/y with x and y replaced appropriately to describe the address and mask length associated with the scope."

::= { ipMRouteScopeNameEntry 4 }

ipMRouteScopeNameDefault OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"If true, indicates a preference that the name in the following language should be used by applications if no name is available in a desired language."

DEFVAL { false }

::= { ipMRouteScopeNameEntry 5 }

ipMRouteScopeNameStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The status of this row, by which new entries may be created, or old entries deleted from this table."

::= { ipMRouteScopeNameEntry 6 }

-- conformance information

ipMRouteMIBConformance

OBJECT IDENTIFIER ::= { ipMRouteStdMIB 2 }

ipMRouteMIBCompliances

OBJECT IDENTIFIER ::= { ipMRouteMIBConformance 1 }

ipMRouteMIBGroups OBJECT IDENTIFIER ::= { ipMRouteMIBConformance 2 }

-- compliance statements

ipMRouteMIBCompliance MODULE-COMPLIANCE

STATUS current

DESCRIPTION

"The compliance statement for the IP Multicast MIB."

MODULE -- this module

MANDATORY-GROUPS { ipMRouteMIBBasicGroup,

```
ipMRouteMIBRouteGroup}
```

```
GROUP ipMRouteMIBBoundaryGroup
```

```
DESCRIPTION
```

```
"This group is mandatory if the router supports
administratively-scoped multicast address boundaries."
```

```
OBJECT ipMRouteBoundaryStatus
```

```
MIN-ACCESS read-only
```

```
DESCRIPTION
```

```
"Write access is not required."
```

```
OBJECT ipMRouteScopeNameStatus
```

```
MIN-ACCESS read-only
```

```
DESCRIPTION
```

```
"Write access is not required."
```

```
GROUP ipMRouteMIBHCInterfaceGroup
```

```
DESCRIPTION
```

```
"This group is mandatory only for those network interfaces
for which the value of the corresponding instance of ifSpeed
is greater than 20,000,000 bits/second."
```

```
::= { ipMRouteMIBCompliances 1 }
```

```
-- units of conformance
```

```
ipMRouteMIBBasicGroup OBJECT-GROUP
```

```
OBJECTS { ipMRouteEnable, ipMRouteEntryCount,
ipMRouteUpstreamNeighbor, ipMRouteInIfIndex,
ipMRouteUpTime, ipMRouteExpiryTime,
ipMRouteNextHopState,
ipMRouteNextHopUpTime,
ipMRouteNextHopExpiryTime,
ipMRouteNextHopProtocol,
ipMRouteNextHopPkts,
ipMRouteInterfaceTtl,
ipMRouteInterfaceProtocol, ipMRouteInterfaceRateLimit,
ipMRouteInterfaceInMcastOctets,
ipMRouteInterfaceOutMcastOctets,
ipMRouteProtocol
}
```

```
STATUS current
```

```
DESCRIPTION
```

```
"A collection of objects to support basic management of IP
Multicast routing."
```

```
::= { ipMRouteMIBGroups 1 }
```

```
ipMRouteMIBHopCountGroup OBJECT-GROUP
  OBJECTS { ipMRouteNextHopClosestMemberHops }
  STATUS current
  DESCRIPTION
    "A collection of objects to support management of the use of
    hop counts in IP Multicast routing."
  ::= { ipMRouteMIBGroups 2 }

ipMRouteMIBBoundaryGroup OBJECT-GROUP
  OBJECTS { ipMRouteBoundaryStatus, ipMRouteScopeNameString,
            ipMRouteScopeNameDefault, ipMRouteScopeNameStatus }
  STATUS current
  DESCRIPTION
    "A collection of objects to support management of scoped
    multicast address boundaries."
  ::= { ipMRouteMIBGroups 3 }

ipMRouteMIBPktsOutGroup OBJECT-GROUP
  OBJECTS { ipMRouteNextHopPkts }
  STATUS current
  DESCRIPTION
    "A collection of objects to support management of packet
    counters for each outgoing interface entry of a route."
  ::= { ipMRouteMIBGroups 4 }

ipMRouteMIBHCInterfaceGroup OBJECT-GROUP
  OBJECTS { ipMRouteInterfaceHCInMcastOctets,
            ipMRouteInterfaceHCOutMcastOctets,
            ipMRouteHCOctets }
  STATUS current
  DESCRIPTION
    "A collection of objects providing information specific to
    high speed (greater than 20,000,000 bits/second) network
    interfaces."
  ::= { ipMRouteMIBGroups 5 }

ipMRouteMIBRouteGroup OBJECT-GROUP
  OBJECTS { ipMRouteRtProto, ipMRouteRtAddress,
            ipMRouteRtMask, ipMRouteRtType }
  STATUS current
  DESCRIPTION
    "A collection of objects providing information on the
    relationship between multicast routing information, and the
    IP Forwarding Table."
  ::= { ipMRouteMIBGroups 6 }

ipMRouteMIBPktsGroup OBJECT-GROUP
  OBJECTS { ipMRoutePkts, ipMRouteDifferentInIfPackets,
```

```
        ipMRouteOctets }
STATUS   current
DESCRIPTION
        "A collection of objects to support management of packet
        counters for each forwarding entry."
 ::= { ipMRouteMIBGroups 7 }
```

END

5. IANA Considerations

The ipMRouteRtProto, ipMRouteNextHopProtocol, ipMRouteInterfaceProtocol, and ipMRouteProtocol use textual conventions imported from the IANA-RTPROTO-MIB. The purpose of defining these textual conventions in a separate MIB module is to allow additional values to be defined without having to issue a new version of this document. The Internet Assigned Numbers Authority (IANA) is responsible for the assignment of all Internet numbers, including various SNMP-related numbers; it will administer the values associated with these textual conventions.

The rules for additions or changes to the IANA-RTPROTO-MIB are outlined in the DESCRIPTION clause associated with its MODULE-IDENTITY statement.

The current versions of the IANA-RTPROTO-MIB can be accessed from the IANA home page at: "<http://www.iana.org/>".

6. Security Considerations

This MIB contains readable objects whose values provide information related to multicast routing, including information on what machines are sending to which groups. There are also a number of objects that have a MAX-ACCESS clause of read-write and/or read-create, such as those which allow an administrator to configure multicast boundaries.

While unauthorized access to the readable objects is relatively innocuous, unauthorized access to the write-able objects could cause a denial of service, or could cause wider distribution of packets intended only for local distribution. Hence, the support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations.

SNMPv1 by itself is such an insecure 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 SET (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 [12] and the View-based Access Control Model RFC 2575 [15] is recommended.

It is then a customer/user responsibility to ensure that the SNMP entity giving access to this MIB, is properly configured to give access to those objects only to those principals (users) that have legitimate rights to access them.

7. Intellectual Property Notice

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

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

8. Acknowledgements

This MIB module was updated based on feedback from the IETF's Inter-Domain Multicast Routing (IDMR) Working Group.

9. Authors' Addresses

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

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

Dino Farinacci
Procket Networks
3850 North First Street
San Jose, CA 95134

Phone: +1 408-954-7909
Email: dino@procket.com

Dave Thaler
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399

Phone: +1 425 703 8835
EMail: dthaler@microsoft.com

10. References

- [1] Wijnen, B., Harrington, D. and R. Presuhn, "An Architecture for Describing SNMP Management Frameworks", RFC 2571, April 1999.
- [2] Rose, M. and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", STD 16, RFC 1155, May 1990.
- [3] Rose, M. and K. McCloghrie, "Concise MIB Definitions", STD 16, RFC 1212, March 1991.
- [4] Rose, M., "A Convention for Defining Traps for use with the SNMP", RFC 1215, March 1991.
- [5] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, STD 58, April 1999.
- [6] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.
- [7] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.
- [8] Case, J., Fedor, M., Schoffstall, M. and J. Davin, "Simple Network Management Protocol", STD 15, RFC 1157, May 1990.
- [9] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Introduction to Community-based SNMPv2", RFC 1901, January 1996.
- [10] 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.
- [11] Case, J., Harrington D., Presuhn R. and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", RFC 2572, April 1999.
- [12] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", RFC 2574, April 1999.

- [13] 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.
- [14] Levi, D., Meyer, P. and B. Stewart, "SNMPv3 Applications", RFC 2573, April 1999.
- [15] Wijnen, B., Presuhn, R. and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", RFC 2575, April 1999.
- [16] Deering, S., "Multicast Routing in a Datagram Internetwork", PhD thesis, Electrical Engineering Dept., Stanford University, December 1991.
- [17] Waitzman, D., Partridge, C. and S. Deering, "Distance Vector Multicast Routing Protocol", RFC 1075, November 1988.
- [18] Estrin, D., Farinacci, D., Helmy, A., Thaler, D., Deering, S., Handley, M., Jacobson, V., Liu, C., Sharma, P. and L. Wei, "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification", RFC 2362, June 1998.
- [19] Deering, S., Estrin, D., Farinacci, D., Jacobson, V., Helmy, A. and L. Wei, "Protocol Independent Multicast Version 2, Dense Mode Specification", Work in Progress.
- [20] Moy, J., "Multicast Extensions to OSPF", RFC 1584, March 1994.
- [21] Ballardie, A., "Core Based Trees (CBT version 2) Multicast Routing", RFC 2189, September 1997.
- [22] Meyer, D., "Administratively Scoped IP Multicast", BCP 23, RFC 2365, July 1998.

11. Full Copyright Statement

Copyright (C) The Internet Society (2000). All Rights Reserved.

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

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

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

Acknowledgement

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

