

Network Working Group
Internet Draft
Expires: October 2005

Susheela Vaidya
Thomas D. Nadeau
Cisco Systems, Inc.
Harmen Van der Linde
AT&T

April 2005

**Multicast in BGP/MPLS IP VPNs
Management Information Base**

[draft-svaidya-mcast-vpn-mib-02.txt](#)

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC 2026](#).

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>.

IPR Disclosure Acknowledgment

By submitting this Internet-Draft, I certify that any applicable patent or other IPR claims of which I am aware have been disclosed, or will be disclosed, and any of which I become aware will be disclosed, in accordance with [RFC 3668](#).

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 to configure and/or monitor multicast in BGP/MPLS IP VPNs as per [\[MCAST-VPN\]](#).

Table of Contents

Abstract	1
1. Introduction	2
2. The SNMP Management Framework	2
3. Assumptions and Prerequisites	2
4. Terminology	3
5. Summary of MCAST VPN MIB Module	4
5.1. mcastVpnGenericTable	4
5.2. mcastVpnMdtDefaultTable	4
5.3. mcastVpnMdtDataTable	4
5.4. mcastVpnMrouteMdtTable	4
5.5. mcastVpnBgpMdtUpdateTable	4
5.6. mcastVpnMdtJnRcvTable	5
5.7. mcastVpnMdtJnSendTable	5
5.8. mcastVpnTunnelTable	5
6. MCAST VPN MIB Module Definitions	5
7. Security Considerations	30
8. IANA Considerations.....	30
9. Acknowledgments	31
10. References	31
10.1. Normative References	31
10.2. Informative References	32
11. Authors' Addresses	32
12. Full Copyright Statement	33
13. Intellectual Property Notice	33

[1.](#) Introduction

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes objects for managing multicast in BGP/MPLS IP VPNs as per [[MCAST-VPN](#)].

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#), reference [[RFC2119](#)].

[2.](#) The SNMP Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to [section 7 of RFC 3410](#) [[RFC3410](#)].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP).

Objects in the MIB are defined using the mechanisms defined in the

Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIV2, which is described in STD 58, [RFC 2578](#) [[RFC2578](#)], STD 58, [RFC 2579](#) [[RFC2579](#)] and STD 58, [RFC 2580](#) [[RFC2580](#)].

3. Assumptions and Prerequisites

Familiarity with the terminology and procedures of [[RFC2547bis](#)], [[MCAST-VPN](#)] and familiarity with IP multicast and PIM is assumed. The managed objects here are closely related to and work along with some of the managed objects defined in [[MPLS-L3VPN-MIB](#)].

4. Terminology

This document uses terminology from the document [[MCAST-VPN](#)] which specifies the necessary protocols and procedures for support of IP multicast in MPLS/BGP IP VPNs.

Some of the terminology is specified here again for an easier read of the document.

MVPN - Multicast in BGP/MPLS IP VPNs.

MVRF - Multicast VRF, a VRF which has multicast enabled.

MDT - Multicast distribution tree.

MD - Multicast Domain, a set of VRFs associated with interfaces that can send multicast traffic to one another.

MDT group - Each MD is assigned a distinct group address from a pool that is administered by the service provider(s). Such groups reserved for MDs are called MDT groups. MDT groups are used to encapsulate and transport multicast VPN traffic within the corresponding MD. MDT group addresses are P-group addresses.

Default MDT/MDT-default - The MDT group used for forwarding MVPN multicast traffic (by default) and for forwarding PIM control traffic.

Data MDT/MDT-data - Range of MDT group addresses that is created on demand for specific MVPN groups. MVPN traffic that is initially forwarded on the default MDT can be moved to a data MDT to minimize flooding the other PE routers in the same MD.

MDT Join TLVs - When a MVPN multicast stream meets the criteria for switching over to data MDT from the default MDT, a PE device picks a

group from the configured data MDT pool and advertises the (C-Source, C-Group, data MDT ie P-Group) mapping to all PE routers in that MD. The advertisements are done via UDP TLV messages and are termed MDT Join TLVs.

MT - Multicast/MVPN Tunnel Interface. For every MD an MVRF is part of, a PE router creates a special interface called the multicast/MVPN tunnel interface. An MD can be thought of as a set of PE routers connected by a multicast tunnel. From the perspective of a VPN-specific PIM instance, an MT is a single multi-access interface.

5. Summary of MIB Module

This MIB enables monitoring of MVRFs, default and data MDTs used in a MD, MDT Join TLVs sent and received for data MDT signalling, distribution of MDT-SAFI NLRI in case of PIM-SSM for discovery of other PE sources in a MD, the MTs set up to connect to MDs etc - all on a PE device. It also enables configuration of MVRFs and MDT addresses. It should be noted that this MIB should be used along with [\[MPLS-L3VPN-MIB\]](#) for a more complete configuration capability. And this MIB, along with the existing standard multicast MIBs provides a more detailed MVPN monitoring capability.

The following subsections describe the purpose of each of the objects contained in the MIB.

5.1. mcastVpnGenericTable

An entry in this table is created for every MVRF in the device.

Note that many implementations may have MVRF for global VRF (VRF0) by default in the device. Also note that existence of the corresponding VRF in [\[MPLS-L3VPN-MIB\]](#) is necessary for a row to exist in this table. Deletion of corresponding VRF in [\[MPLS-L3VPN-MIB\]](#) also results in deletion of a row here.

5.2. mcastVpnMdtDefaultTable

This table contains objects that show the default MDT address for a given MVRF and the encapsulation type used.

5.3. mcastVpnMdtDataTable

This table contains objects that show the data MDT address pool for a given MVRF. It is recommended that the implementations make sure an entry in mcastVpnMdtDefaultTable for a MVRF be present before an entry for that MVRF is created in mcastVpnMdtDataTable.

5.4. mcastVpnMrouteMdtTable

This table provides information about the MDT group being used at the instance of querying for a given multicast traffic stream in an MD. This table is the result of static configurations of default and/or data MDT groups and dynamic operational conditions (usually bandwidth threshold) chosen by the implementation.

5.5. mcastVpnBgpMdtUpdateTable

As described in [[MCAST-VPN](#)], in the case of PIM-SSM, the necessary MDTs for an MD cannot be set up until each PE in the MD knows the source address of each of the other PEs in that MD. To facilitate the auto-discovery of this information, a new BGP Address Family is defined, the NLRI for which consists of originating PE's address (source address used over the MDT in question), P-group address to be used as the default MDT address in the MDT in question amongst other things. When a PE receives such an MDT-SAFI NLRI or simply put - a BGP MDT update or advertisement, it caches this information and uses it to join the MDT. The mcastVpnBgpMdtUpdateTable provides this received and cached information in a PE.

5.6. mcastVpnMdtJnRcvTable

This table has information about the MDT Join TLVs received by a PE.

5.7. mcastVpnMdtJnSendTable

This table has information about the MDT Join TLVs being sent by a PE.

5.8. mcastVpnTunnelTable

This table provides the list of MVPN tunnels (MTs) present in a PE, the MVRFs they are associated with and any other MVPN technology specific information about these tunnels. Note that an MT is a regular tunnel interface with ifType 'tunnel' (131) and is supposed to have generic interface specific support in the IF-MIB and generic tunnel support in the [RFC2667](#).

6.0 MIB Module Definitions

```
MCAST-VPN-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE, Unsigned32
FROM SNMPv2-SMI
```

```
MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
FROM SNMPv2-CONF
```

```
TruthValue, RowStatus, TimeStamp, DisplayString, TimeInterval
```


FROM SNMPv2-TC

ifIndex

FROM IF-MIB

SnmpAdminString

FROM SNMP-FRAMEWORK-MIB

InetAddress, InetAddressType

FROM INET-ADDRESS-MIB

mplsVpnVrfName, MplsVpnRouteDistinguisher

FROM MPLS-L3VPN-MIB;

mcastVpnMIB MODULE-IDENTITY

LAST-UPDATED "200407011200Z" -- 01 July 2004 12:00:00 GMT

ORGANIZATION "IETF Layer-3 Virtual Private
Networks Working Group."

CONTACT-INFO

" Susheela Vaidya
svaidya@cisco.com

Thomas Nadeau
tnadeau@cisco.com

Harmen Van der Linde
hvd1@att.com

Comments and discussion to l3vpn@ietf.org"

DESCRIPTION

"This MIB contains managed object definitions for
multicast in BGP/MPLS IP VPNs defined
by [[MCAST-VPN](#)].
Copyright (C) The Internet Society (2004)."

-- Revision history.

REVISION "200407011200Z" -- 01 July 2004 12:00:00 GMT

DESCRIPTION

"Initial version."

::= { experimental xyz }

-- Top level components of this MIB.

mcastVpnNotifications OBJECT IDENTIFIER ::= { mcastVpnMIB 0 }

mcastVpnObjects OBJECT IDENTIFIER ::= { mcastVpnMIB 1 }

mcastVpnScalars OBJECT IDENTIFIER ::= { mcastVpnObjects 1 }

mcastVpnGeneric OBJECT IDENTIFIER ::= { mcastVpnObjects 2 }

mcastVpnConfig OBJECT IDENTIFIER ::= { mcastVpnObjects 3 }

mcastVpnProtocol OBJECT IDENTIFIER ::= { mcastVpnObjects 4 }


```
mcastVpnConformance    OBJECT IDENTIFIER ::= { mcastVpnMIB 2 }
```

```
-- Scalar Objects
```

```
mcastVpnMvrfNumber OBJECT-TYPE
```

```
    SYNTAX      Unsigned32
```

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

```
    STATUS      current
```

```
    DESCRIPTION
```

```
        "The number of MVRFs that are present in this device."
```

```
    ::= { mcastVpnScalars 1 }
```

```
mcastVpnNotificationEnable OBJECT-TYPE
```

```
    SYNTAX      TruthValue
```

```
    MAX-ACCESS  read-write
```

```
    STATUS      current
```

```
    DESCRIPTION
```

```
        "If this object is TRUE, then the generation of all  
        notifications defined in this MIB is enabled."
```

```
    DEFVAL { false }
```

```
    ::= { mcastVpnScalars 2 }
```

```
-- Generic MVRF Information Table
```

```
mcastVpnGenericTable OBJECT-TYPE
```

```
    SYNTAX      SEQUENCE OF McastVpnGenericEntry
```

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

```
    STATUS      current
```

```
    DESCRIPTION
```

```
        "This table gives the generic information about the MVRFs  
        present in this device."
```

```
    ::= { mcastVpnGeneric 1 }
```

```
mcastVpnGenericEntry OBJECT-TYPE
```

```
    SYNTAX      McastVpnGenericEntry
```

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

```
    STATUS      current
```

```
    DESCRIPTION
```

```
        "An entry in this table is created for every MVRF in the  
        device."
```

```
    INDEX      { mplsVpnVrfName }
```

```
    ::= { mcastVpnGenericTable 1 }
```

```
McastVpnGenericEntry ::= SEQUENCE {
```

```
    mcastVpnGenOperStatusChange    INTEGER,
```

```
    mcastVpnGenOperChangeTime      TimeStamp,
```

```
    mcastVpnGenAssociatedInterfaces Unsigned32,
```

```
    mcastVpnGenRowStatus           RowStatus
```

}

mcastVpnGenOperStatusChange OBJECT-TYPE

Vaidya, et al.

Expires October 2005

[Page 7]

SYNTAX INTEGER { createdMvrf(1),
 deletedMvrf(2),
 modifiedMvrfDefMdtConfig(3),
 modifiedMvrfDataMdtConfig(4)
 }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object describes the last operational change that happened for the given MVRF.

createdMvrf - indicates that the MVRF was created in the device.

deletedMvrf - indicates that the MVRF was deleted from the device. A row in this table will never have mcastVpnGenOperStatusChange equal to deletedMvrf(2), because in that case the row itself will be deleted from the table. This value for mcastVpnGenOperStatusChange is defined mainly for use in mcastVpnMvrfChange notification.

modifiedMvrfDefMdtConfig - indicates that the default MDT group for the MVRF was configured, deleted or changed.

modifiedMvrfDataMdtConfig - indicates that the data MDT group range or a associated variable (like the threshold) for the MVRF was configured, deleted or changed."

DEFVAL { createdMvrf }

::= { mcastVpnGenericEntry 1 }

mcastVpnGenOperChangeTime OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The time at which the last operational change for the MVRF in question took place. The last operational change is specified by mcastVpnGenOperStatusChange."

::= { mcastVpnGenericEntry 2 }

mcastVpnGenAssociatedInterfaces OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Total number of interfaces associated with this MVRF (including


```
    the MDT tunnel interface) with ifOperStatus = up(1)."  
 ::= { mcastVpnGenericEntry 3 }
```

mcastVpnGenRowStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This variable is used to create or delete a row in this table."

```
 ::= { mcastVpnGenericEntry 4 }
```

-- MDT-default group Configuration Table

mcastVpnMdtDefaultTable OBJECT-TYPE

SYNTAX SEQUENCE OF McastVpnMdtDefaultEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table specifies the default MDT address and the encapsulation type used for a MVRF instance."

```
 ::= { mcastVpnConfig 1 }
```

mcastVpnMdtDefaultEntry OBJECT-TYPE

SYNTAX McastVpnMdtDefaultEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in this table is created for every MVRF for which a default MDT group is configured. A MVRF which does not have a default MDT group configured will not appear in this table. Creation of a row in this table is the equivalent of configuring default MDT address for the given MVRF. Deletion of a row in this table is the equivalent of deconfiguring default MDT address for the given MVRF."

INDEX { mplsVpnVrfName }

```
 ::= { mcastVpnMdtDefaultTable 1 }
```

McastVpnMdtDefaultEntry ::= SEQUENCE {

mcastVpnMdtDefaultAddrType InetAddressType,

mcastVpnMdtDefaultAddress InetAddress,

mcastVpnMdtEncapsType INTEGER,

mcastVpnMdtDefaultRowStatus RowStatus

}

mcastVpnMdtDefaultAddrType OBJECT-TYPE

SYNTAX InetAddressType

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The Internet address type of mcastVpnMdtDefaultAddress."

::= { mcastVpnMdtDefaultEntry 1 }

mcastVpnMdtDefaultAddress OBJECT-TYPE

SYNTAX InetAddress

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The default MDT address to be used for the MVRF in question."

::= { mcastVpnMdtDefaultEntry 2 }

mcastVpnMdtEncapsType OBJECT-TYPE

SYNTAX INTEGER { greIp (1),
ipIp (2),
mpls (3)
}

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The encapsulation type to be used in the MVRF in question."

::= { mcastVpnMdtDefaultEntry 3 }

mcastVpnMdtDefaultRowStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This variable is used to create, modify or delete a row in
this table."

::= { mcastVpnMdtDefaultEntry 4 }

-- MDT-data configuration table

mcastVpnMdtDataTable OBJECT-TYPE

SYNTAX SEQUENCE OF McastVpnMdtDataEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table specifies the range of data MDT addresses and
associated variables for a MVRF instance."

::= { mcastVpnConfig 2 }

mcastVpnMdtDataEntry OBJECT-TYPE

SYNTAX McastVpnMdtDataEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in this table is created for every MVRF for which a data MDT group range is configured. A MVRF which does not have a data MDT group range configured will not appear in this table.

Creation of a row in this table is the equivalent of configuring data MDT addresses for the given MVRF. Deletion of a row in this table is the equivalent of deconfiguring data MDT address usage in the given MVRF."

INDEX { mplsVpnVrfName }

::= { mcastVpnMdtDataTable 1 }

```
McastVpnMdtDataEntry ::= SEQUENCE {  
    mcastVpnMdtDataRangeAddrType   InetAddressType,  
    mcastVpnMdtDataRangeAddress     InetAddress,  
    mcastVpnMdtDataWildcardType     InetAddressType,  
    mcastVpnMdtDataWildcardBits     InetAddress,  
    mcastVpnMdtDataThreshold        Unsigned32,  
    mcastVpnMdtDataRowStatus        RowStatus  
}
```

mcastVpnMdtDataRangeAddrType OBJECT-TYPE

SYNTAX InetAddressType

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The Internet address type of mcastVpnMdtDataRangeAddress."

::= { mcastVpnMdtDataEntry 1 }

mcastVpnMdtDataRangeAddress OBJECT-TYPE

SYNTAX InetAddress

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The data MDT group range address for the given MVRF. This along with mcastVpnMdtDataWildcardBits gives the pool of data MDT addresses that can be used for encapsulation in the MVRF upon data MDT switchover."

::= { mcastVpnMdtDataEntry 2 }

mcastVpnMdtDataWildcardType OBJECT-TYPE

SYNTAX InetAddressType

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The Internet address type of mcastVpnMdtDataWildcardBits."

::= { mcastVpnMdtDataEntry 3 }

mcastVpnMdtDataWildcardBits OBJECT-TYPE

SYNTAX InetAddress

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Wildcard bits which when used along with data MDT range address, give a pool of addresses to be used in a MVRF.

For example, if mcastVpnMdtDataRangeAddress is 239.1.2.0 and mcastVpnMdtDataWildcardBits is 0.0.0.3, the possible data MDT addresses are 239.1.2.0, 239.1.2.1, 239.1.2.2 and 239.1.2.3.

Note that wild card bits should be right contiguous."

::= { mcastVpnMdtDataEntry 4 }

mcastVpnMdtDataThreshold OBJECT-TYPE

SYNTAX Unsigned32 (0..4294967295)

UNITS "kilobits per second"

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The bandwidth threshold value which when exceeded for a multicast routing entry in the given MVRF, triggers usage of data MDT address instead of default MDT address for encapsulation."

::= { mcastVpnMdtDataEntry 5 }

mcastVpnMdtDataRowStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This variable is used to create, modify or delete a row in this table."

::= { mcastVpnMdtDataEntry 6 }

-- MDT group info for a multicast route entry in question

mcastVpnMrouteMdtTable OBJECT-TYPE

SYNTAX SEQUENCE OF McastVpnMrouteMdtEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Given a multicast routing entry and the context MVRF, this table provides information about the MDT group being used for encapsulating the traffic for the multicast routing entry in the provider network at the instance of querying. Note that this table is a read-only table and is the result of the default MDT and data MDT configurations and the operational conditions like the traffic rate and sometimes, the implementation choices."

::= { mcastVpnProtocol 1 }

mcastVpnMrouteMdtEntry OBJECT-TYPE

SYNTAX McastVpnMrouteMdtEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in this table exists for a multicast routing entry the traffic for which is being encapsulated in a context MVRF."

INDEX { mplsVpnVrfName,
mcastVpnMrouteMvrfGrpAddrType,
mcastVpnMrouteMvrfGroup,
mcastVpnMrouteMvrfSrcAddrType,
mcastVpnMrouteMvrfSource,
mcastVpnMrouteUpDownStreamInfo }

::= { mcastVpnMrouteMdtTable 1 }

McastVpnMrouteMdtEntry ::= SEQUENCE {

mcastVpnMrouteMvrfGrpAddrType	InetAddressType,
mcastVpnMrouteMvrfGroup	InetAddress,
mcastVpnMrouteMvrfSrcAddrType	InetAddressType,
mcastVpnMrouteMvrfSource	InetAddress,
mcastVpnMrouteUpDownStreamInfo	INTEGER,
mcastVpnMrouteMdtGrpAddrType	InetAddressType,
mcastVpnMrouteMdtGroup	InetAddress,
mcastVpnMrouteMdtType	INTEGER

}

mcastVpnMrouteMvrfGrpAddrType OBJECT-TYPE

SYNTAX InetAddressType

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The Internet address type of mcastVpnMrouteMvrfGroup."


```
::= { mcastVpnMrouteMdtEntry 1 }
```

mcastVpnMrouteMvrfGroup OBJECT-TYPE

SYNTAX InetAddress (SIZE (4|16|20))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Group address of multicast routing entry in question."

```
::= { mcastVpnMrouteMdtEntry 2 }
```

mcastVpnMrouteMvrfSrcAddrType OBJECT-TYPE

SYNTAX InetAddressType

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The Internet address type of mcastVpnMrouteMvrfSource."

```
::= { mcastVpnMrouteMdtEntry 3 }
```

mcastVpnMrouteMvrfSource OBJECT-TYPE

SYNTAX InetAddress (SIZE (4|16|20))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Source address of the multicast routing entry in question."

```
::= { mcastVpnMrouteMdtEntry 4 }
```

mcastVpnMrouteUpDownStreamInfo OBJECT-TYPE

SYNTAX INTEGER { upstream (1),
downstream (2)
}

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Indicates if this PE is the upstream (sending or ingress) or the downstream (receiving or egress) router for the multicast routing entry specified by mcastVpnMrouteMvrfSource and mcastVpnMrouteMvrfGroup in the context MVRP specified by mplsVpnVrfName."

```
::= { mcastVpnMrouteMdtEntry 5 }
```

mcastVpnMrouteMdtGrpAddrType OBJECT-TYPE

SYNTAX InetAddressType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The Internet address type of mcastVpnMrouteMdtGroup."


```
::= { mcastVpnMrouteMdtEntry 6 }
```

mcastVpnMrouteMdtGroup OBJECT-TYPE

SYNTAX InetAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"MDT group address used to encapsulate the multicast routing entry specified by mcastVpnMrouteMvrfSource and mcastVpnMrouteMvrfGroup in the context MVRF specified by mplsVpnVrfName."

```
::= { mcastVpnMrouteMdtEntry 7 }
```

mcastVpnMrouteMdtType OBJECT-TYPE

SYNTAX INTEGER { mdtDefault (1),
mdtData (2)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates the type of MDT group used for encapsulation."

```
::= { mcastVpnMrouteMdtEntry 8 }
```

-- Table of BGP MDT Updates received

mcastVpnBgpMdtUpdateTable OBJECT-TYPE

SYNTAX SEQUENCE OF McastVpnBgpMdtUpdateEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table has information about the BGP advertisement of the the MDT groups."

```
::= { mcastVpnProtocol 2 }
```

mcastVpnBgpMdtUpdateEntry OBJECT-TYPE

SYNTAX McastVpnBgpMdtUpdateEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in this table is created when a BGP advertisement of the MDT group is received and cached in the PE device.

An entry in this table deleted when such a cached BGP MDT update is withdrawn."

INDEX { mcastVpnBgpMdtUpdGrpAddrType,
mcastVpnBgpMdtUpdateGroup,
mcastVpnBgpMdtUpdSrcAddrType,


```
        mcastVpnBgpMdtUpdateSource }  
 ::= { mcastVpnBgpMdtUpdateTable 1 }
```

```
McastVpnBgpMdtUpdateEntry ::= SEQUENCE {  
    mcastVpnBgpMdtUpdGrpAddrType      InetAddressType,  
    mcastVpnBgpMdtUpdateGroup          InetAddress,  
    mcastVpnBgpMdtUpdateRd             MplsVpnRouteDistinguisher,  
    mcastVpnBgpMdtUpdSrcAddrType       InetAddressType,  
    mcastVpnBgpMdtUpdateSource         InetAddress,  
    mcastVpnBgpMdtUpdOrigAddrType      InetAddressType,  
    mcastVpnBgpMdtUpdateOriginator     InetAddress,  
    mcastVpnBgpMdtUpdNhAddrType        InetAddressType,  
    mcastVpnBgpMdtUpdateNextHop        InetAddress  
}
```

mcastVpnBgpMdtUpdGrpAddrType OBJECT-TYPE

SYNTAX InetAddressType

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

 "The Internet address type of mcastVpnBgpMdtUpdateGroup."

```
 ::= { mcastVpnBgpMdtUpdateEntry 1 }
```

mcastVpnBgpMdtUpdateGroup OBJECT-TYPE

SYNTAX InetAddress (SIZE (4|16|20))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

 "MDT group address in the BGP MDT advertisement."

```
 ::= { mcastVpnBgpMdtUpdateEntry 2 }
```

mcastVpnBgpMdtUpdateRd OBJECT-TYPE

SYNTAX MplsVpnRouteDistinguisher

MAX-ACCESS read-only

STATUS current

DESCRIPTION

 "RD (route distinguisher) in the BGP MDT advertisement. This
 is the RD corresponding to the originator PE."

```
 ::= { mcastVpnBgpMdtUpdateEntry 3 }
```

mcastVpnBgpMdtUpdSrcAddrType OBJECT-TYPE

SYNTAX InetAddressType

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

 "The Internet address type of mcastVpnBgpMdtUpdateSource."


```
::= { mcastVpnBgpMdtUpdateEntry 4 }
```

mcastVpnBgpMdtUpdateSource OBJECT-TYPE

SYNTAX InetAddress (SIZE (4|16|20))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"MDT source address in the BGP MDT advertisement."

```
::= { mcastVpnBgpMdtUpdateEntry 5 }
```

mcastVpnBgpMdtUpdOrigAddrType OBJECT-TYPE

SYNTAX InetAddressType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The Internet address type of mcastVpnBgpMdtUpdateOriginator."

```
::= { mcastVpnBgpMdtUpdateEntry 6 }
```

mcastVpnBgpMdtUpdateOriginator OBJECT-TYPE

SYNTAX InetAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The BGP peering address of the device that originated (or advertised) the BGP MDT update."

```
::= { mcastVpnBgpMdtUpdateEntry 7 }
```

mcastVpnBgpMdtUpdNhAddrType OBJECT-TYPE

SYNTAX InetAddressType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The Internet address type of mcastVpnBgpMdtUpdateNexthop."

```
::= { mcastVpnBgpMdtUpdateEntry 8 }
```

mcastVpnBgpMdtUpdateNexthop OBJECT-TYPE

SYNTAX InetAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The next-hop address (address of the border router to be used to reach the destination network) in the BGP MDT advertisement."

```
::= { mcastVpnBgpMdtUpdateEntry 9 }
```

-- Table of MDT-data joins received

mcastVpnMdtJnRcvTable OBJECT-TYPE

SYNTAX SEQUENCE OF McastVpnMdtJnRcvEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table has information about the data MDT join TLVs received by a device."

::= { mcastVpnProtocol 3 }

mcastVpnMdtJnRcvEntry OBJECT-TYPE

SYNTAX McastVpnMdtJnRcvEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in this table is created or updated for every MDT data join TLV received and cached in the device. The value of mplsVpnVrfName in such an entry specifies the name of the MVRF for which the data MDT groups from the TLVs are used."

INDEX { mplsVpnVrfName,
mcastVpnMdtJnRcvGrpAddrType,
mcastVpnMdtJnRcvGroup,
mcastVpnMdtJnRcvSrcAddrType,
mcastVpnMdtJnRcvSource }

::= { mcastVpnMdtJnRcvTable 1 }

McastVpnMdtJnRcvEntry ::= SEQUENCE {

mcastVpnMdtJnRcvGrpAddrType InetAddressType,

mcastVpnMdtJnRcvGroup InetAddress,

mcastVpnMdtJnRcvSrcAddrType InetAddressType,

mcastVpnMdtJnRcvSource InetAddress,

mcastVpnMdtJnRcvUpTime TimeInterval,

mcastVpnMdtJnRcvExpTime TimeInterval

}

mcastVpnMdtJnRcvGrpAddrType OBJECT-TYPE

SYNTAX InetAddressType

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The Internet address type of mcastVpnMdtJnRcvGroup."

::= { mcastVpnMdtJnRcvEntry 1 }

mcastVpnMdtJnRcvGroup OBJECT-TYPE

SYNTAX InetAddress (SIZE (4|16|20))

MAX-ACCESS not-accessible

STATUS current
DESCRIPTION
 "Data MDT group address in the MDT join TLV."
 ::= { mcastVpnMdtJnRcvEntry 2 }

mcastVpnMdtJnRcvSrcAddrType OBJECT-TYPE

SYNTAX InetAddressType
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
 "The Internet address type of mcastVpnMdtJnRcvSource."
 ::= { mcastVpnMdtJnRcvEntry 3 }

mcastVpnMdtJnRcvSource OBJECT-TYPE

SYNTAX InetAddress (SIZE (4|16|20))
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
 "Source address for the MDT multicast routing entry created
 following the receipt of MDT join TLV."
 ::= { mcastVpnMdtJnRcvEntry 4 }

mcastVpnMdtJnRcvUpTime OBJECT-TYPE

SYNTAX TimeInterval
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "The time since this MDT join TLV was first received by the
 device."
 ::= { mcastVpnMdtJnRcvEntry 5 }

mcastVpnMdtJnRcvExpTime OBJECT-TYPE

SYNTAX TimeInterval
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "The amount of time remaining before the cache corresponding
 to this MDT join TLV is deleted from the device and the
 corresponding MDT multicast routing entry is marked as a
 non-MDT entry.
 Note that multiple TLVs for a data MDT group may be received
 by a device. Upon receipt, the expiry timer of an already
 existing entry is restarted and so mcastVpnMdtJnRcvExpTime
 is updated."
 ::= { mcastVpnMdtJnRcvEntry 6 }

-- Table for MDT-data join TLVs sent

mcastVpnMdtJnSendTable OBJECT-TYPE

SYNTAX SEQUENCE OF McastVpnMdtJnSendEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table specifies the data MDT Join TLVs sent by a device."

::= { mcastVpnProtocol 4 }

mcastVpnMdtJnSendEntry OBJECT-TYPE

SYNTAX McastVpnMdtJnSendEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Entries in this table exist for data MDT Join TLVs that are being sent by this device to other PEs."

INDEX { mplsVpnVrfName,
mcastVpnMdtJnSendGrpAddrType,
mcastVpnMdtJnSendGroup,
mcastVpnMdtJnSendSrcAddrType,
mcastVpnMdtJnSendSource }

::= { mcastVpnMdtJnSendTable 1 }

McastVpnMdtJnSendEntry ::= SEQUENCE {

mcastVpnMdtJnSendGrpAddrType InetAddressType,

mcastVpnMdtJnSendGroup InetAddress,

mcastVpnMdtJnSendSrcAddrType InetAddressType,

mcastVpnMdtJnSendSource InetAddress,

mcastVpnMdtJnSendMdtGrpAddrType InetAddressType,

mcastVpnMdtJnSendMdtGroup InetAddress,

mcastVpnMdtJnSendMdtRefCt Unsigned32

}

mcastVpnMdtJnSendGrpAddrType OBJECT-TYPE

SYNTAX InetAddressType

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The Internet address type of mcastVpnMdtJnSendGroup."

::= { mcastVpnMdtJnSendEntry 1 }

mcastVpnMdtJnSendGroup OBJECT-TYPE

SYNTAX InetAddress (SIZE (4|16|20))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This indicates the address of a multicast group in the MVRF specified by the column mplsVpnVrfName. This along with mcastVpnMdtJnSendSource identifies the multicast routing entry for which the MDT join TLV is sent."

::= { mcastVpnMdtJnSendEntry 2 }

mcastVpnMdtJnSendSrcAddrType OBJECT-TYPE

SYNTAX InetAddressType

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The Internet address type of mcastVpnMdtJnSendSource."

::= { mcastVpnMdtJnSendEntry 3 }

mcastVpnMdtJnSendSource OBJECT-TYPE

SYNTAX InetAddress (SIZE (4|16|20))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This indicates the address of a source in the MVRF specified by the column mplsVpnVrfName. This, along with mcastVpnMdtJnSendGroup identifies the multicast routing entry for which the MDT join TLV is sent."

::= { mcastVpnMdtJnSendEntry 4 }

mcastVpnMdtJnSendMdtGrpAddrType OBJECT-TYPE

SYNTAX InetAddressType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The Internet address type of mcastVpnMdtJnSendMdtGroup."

::= { mcastVpnMdtJnSendEntry 5 }

mcastVpnMdtJnSendMdtGroup OBJECT-TYPE

SYNTAX InetAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The data MDT group in the MDT Join TLV sent."

::= { mcastVpnMdtJnSendEntry 6 }

mcastVpnMdtJnSendMdtRefCt OBJECT-TYPE

SYNTAX Unsigned32 (0..65535)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates how many multicast routing entries in the MVRF specified by the column mplsVpnVrfName are using mcastVpnMdtJnSendMdtGroup for encapsulation."

::= { mcastVpnMdtJnSendEntry 7 }

-- Table of MVPN tunnel interfaces

mcastVpnTunnelTable OBJECT-TYPE

SYNTAX SEQUENCE OF McastVpnTunnelEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table gives information about the MVPN/MDT tunnels present in the device."

::= { mcastVpnProtocol 5 }

mcastVpnTunnelEntry OBJECT-TYPE

SYNTAX McastVpnTunnelEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in this table is created for every MVPN tunnel interface present in the device. The ifType for a MVPN tunnel is 'tunnel' (131)."

INDEX { ifIndex }

::= { mcastVpnTunnelTable 1 }

McastVpnTunnelEntry ::= SEQUENCE {

mcastVpnTunnelName DisplayString,

mcastVpnTunnelMvrf SnmpAdminString

}

mcastVpnTunnelName OBJECT-TYPE

SYNTAX DisplayString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The canonical name assigned to the tunnel. The ifName of this tunnel interface should have a value equal to mcastVpnTunnelName."

::= { mcastVpnTunnelEntry 1 }

mcastVpnTunnelMvrf OBJECT-TYPE

SYNTAX SnmpAdminString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Name of the MVRF that this tunnel is associated with.
This object has the same value as mplsVpnVrfName
for the MVRF."

::= { mcastVpnTunnelEntry 2 }

-- MVPN Notifications

mcastVpnMvrfChange NOTIFICATION-TYPE

OBJECTS {
mcastVpnGenOperStatusChange
}

STATUS current

DESCRIPTION

"A mcastVpnMvrfChange notification signifies a change about a MVRF in the device. The change event can be creation of the MVRF, deletion of the MVRF or an update on the default or data MDT configuration of the MVRF. The change event is indicated by mcastVpnGenOperStatusChange embedded in the notification. The user can then query mcastVpnGenericTable, mcastVpnMdtDefaultTable and/or mcastVpnMdtDataTable to get the details of the change as necessary.

Note: Since the creation of a MVRF is often followed by configuration of default and data MDT groups for the MVRF, more than one (three at most) notifications for a MVRF may be generated serially, and it is really not necessary to generate all three of them. An agent may choose to generate a notification for the last event only, that is for data MDT configuration.

Similarly, deletion of default or data MDT configuration on a MVRF happens before a MVRF is deleted and it is recommended that the agent send the notification for MVRF deletion event only."

::= { mcastVpnNotifications 2 }

-- Conformance information

mcastVpnGroups

OBJECT IDENTIFIER ::= { mcastVpnConformance 1 }

mcastVpnCompliances

OBJECT IDENTIFIER ::= { mcastVpnConformance 2 }

-- Module Compliance

mcastVpnModuleFullCompliance MODULE-COMPLIANCE

STATUS current

DESCRIPTION

"Compliance statement for agents that support read-create
so that both configuration and monitoring can be
accomplished via this MIB module."

MODULE -- this module

MANDATORY-GROUPS { mcastVpnMIBScalarGroup,
mcastVpnMIBGenericGroup,
mcastVpnMIBMdtDefaultGroup,
mcastVpnMIBMdtDataGroup,
mcastVpnMIBMrouteMdtGroup,
mcastVpnMIBBgpMdtUpdateGroup,
mcastVpnMIBMdtJnRcvGroup,
mcastVpnMIBMdtJnSendGroup,
mcastVpnMIBTunnelGroup,
mcastVpnMIBNotificationGroup
}

OBJECT mcastVpnGenRowStatus

SYNTAX RowStatus

DESCRIPTION

"Support for notReady(3) and createAndWait(5) is not
required."

OBJECT mcastVpnMdtDefaultAddrType

DESCRIPTION

"Not all Internet address types need to be supported."

OBJECT mcastVpnMdtEncapsType

DESCRIPTION

"Not all encapsulation types defined need to be supported.
Only one type may be supported by default."

OBJECT mcastVpnMdtDefaultRowStatus

SYNTAX RowStatus

DESCRIPTION

"Support for notReady(3) and createAndWait(5) is not
required."

OBJECT mcastVpnMdtDataRangeAddrType

DESCRIPTION

"Not all Internet address types need to be supported."

OBJECT mcastVpnMdtDataWildcardType

DESCRIPTION

"Not all Internet address types need to be supported."

OBJECT mcastVpnMdtDataRowStatus

SYNTAX RowStatus

DESCRIPTION

"Support for notReady(3) and createAndWait(5) is not required."

OBJECT mcastVpnMrouteMdtGrpAddrType

DESCRIPTION

"Not all Internet address types need to be supported."

OBJECT mcastVpnBgpMdtUpdOrigAddrType

DESCRIPTION

"Not all Internet address types need to be supported."

OBJECT mcastVpnBgpMdtUpdNhAddrType

DESCRIPTION

"Not all Internet address types need to be supported."

OBJECT mcastVpnMdtJnSendMdtGrpAddrType

DESCRIPTION

"Not all Internet address types need to be supported."

::= { mcastVpnCompliances 1 }

mcastVpnModuleReadOnlyCompliance MODULE-COMPLIANCE

STATUS current

DESCRIPTION

"Compliance statement for agents that support the monitoring of MVRFs via this MIB module."

MODULE -- this module

MANDATORY-GROUPS { mcastVpnMIBScalarGroup,
mcastVpnMIBGenericGroup,
mcastVpnMIBMdtDefaultGroup,
mcastVpnMIBMdtDataGroup,
mcastVpnMIBMrouteMdtGroup,
mcastVpnMIBBgpMdtUpdateGroup,
mcastVpnMIBMdtJnRcvGroup,


```
        mcastVpnMIBMdtJnSendGroup,  
        mcastVpnMIBTunnelGroup,  
        mcastVpnMIBNotificationGroup  
    }
```

OBJECT mcastVpnGenRowStatus

SYNTAX RowStatus

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required and active(1) is the only
status that needs to be supported."

OBJECT mcastVpnMdtDefaultAddrType

MIN-ACCESS read-only

DESCRIPTION

"Not all Internet address types need to be supported. Write
access is not required."

OBJECT mcastVpnMdtDefaultAddress

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT mcastVpnMdtEncapsType

MIN-ACCESS read-only

DESCRIPTION

"Not all encapsulation types defined need to be supported.
Only one type may be supported by default. Write access
is not required."

OBJECT mcastVpnMdtDefaultRowStatus

SYNTAX RowStatus

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required and active(1) is the only
status that needs to be supported."

OBJECT mcastVpnMdtDataRangeAddrType

MIN-ACCESS read-only

DESCRIPTION

"Not all Internet address types need to be supported.
Write access is not required."

OBJECT mcastVpnMdtDataRangeAddress

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT mcastVpnMdtDataWildcardType

MIN-ACCESS read-only

DESCRIPTION

"Not all Internet address types need to be supported.
Write access is not required."

OBJECT mcastVpnMdtDataWildcardBits

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT mcastVpnMdtDataThreshold

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT mcastVpnMdtDataRowStatus

SYNTAX RowStatus

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required and active(1) is the only
status that needs to be supported."

OBJECT mcastVpnMrouteMdtGrpAddrType

DESCRIPTION

"Not all Internet address types need to be supported."

OBJECT mcastVpnBgpMdtUpdOrigAddrType

DESCRIPTION

"Not all Internet address types need to be supported."

OBJECT mcastVpnBgpMdtUpdNhAddrType

DESCRIPTION

"Not all Internet address types need to be supported."

OBJECT mcastVpnMdtJnSendMdtGrpAddrType

DESCRIPTION

"Not all Internet address types need to be supported."

::= { mcastVpnCompliances 2 }

-- Units of conformance.

mcastVpnMIBScalarGroup OBJECT-GROUP


```
OBJECTS { mcastVpnMvrfNumber,
          mcastVpnNotificationEnable
        }
STATUS   current
DESCRIPTION
    "Collection of scalar objects required for MVPN management."
 ::= { mcastVpnGroups 1 }
```

```
mcastVpnMIBGenericGroup OBJECT-GROUP
OBJECTS { mcastVpnGenOperStatusChange,
          mcastVpnGenOperChangeTime,
          mcastVpnGenAssociatedInterfaces,
          mcastVpnGenRowStatus
        }
STATUS   current
DESCRIPTION
    "Collection of objects needed for MVPN MVRF management."
 ::= { mcastVpnGroups 2 }
```

```
mcastVpnMIBMdtDefaultGroup OBJECT-GROUP
OBJECTS { mcastVpnMdtDefaultAddrType,
          mcastVpnMdtDefaultAddress,
          mcastVpnMdtEncapsType,
          mcastVpnMdtDefaultRowStatus
        }
STATUS   current
DESCRIPTION
    "Collection of objects needed for MVPN default MDT group
    management."
 ::= { mcastVpnGroups 3 }
```

```
mcastVpnMIBMdtDataGroup OBJECT-GROUP
OBJECTS { mcastVpnMdtDataRangeAddrType,
          mcastVpnMdtDataRangeAddress,
          mcastVpnMdtDataWildcardType,
          mcastVpnMdtDataWildcardBits,
          mcastVpnMdtDataThreshold,
          mcastVpnMdtDataRowStatus
        }
STATUS   current
DESCRIPTION
    "Collection of objects needed for MVPN data MDT group
    management."
 ::= { mcastVpnGroups 4 }
```

```
mcastVpnMIBMrouteMdtGroup OBJECT-GROUP
```



```
OBJECTS { mcastVpnMrouteMdtGrpAddrType,
          mcastVpnMrouteMdtGroup,
          mcastVpnMrouteMdtType
        }
STATUS   current
DESCRIPTION
    "Collection of objects that list sources in a MVRF and
    the corresponding MDT groups."
 ::= { mcastVpnGroups 5 }

mcastVpnMIBBgpMdtUpdateGroup OBJECT-GROUP
OBJECTS { mcastVpnBgpMdtUpdateRd,
          mcastVpnBgpMdtUpdOrigAddrType,
          mcastVpnBgpMdtUpdateOriginator,
          mcastVpnBgpMdtUpdNhAddrType,
          mcastVpnBgpMdtUpdateNexthop
        }
STATUS   current
DESCRIPTION
    "Collection of objects needed for MVPN BGP MDT updates
    related information."
 ::= { mcastVpnGroups 6 }

mcastVpnMIBMdtJnRcvGroup OBJECT-GROUP
OBJECTS { mcastVpnMdtJnRcvUpTime,
          mcastVpnMdtJnRcvExpTime
        }
STATUS   current
DESCRIPTION
    "Collection of objects needed for management of MVPN MDT
    data joins received by a device."
 ::= { mcastVpnGroups 7 }

mcastVpnMIBMdtJnSendGroup OBJECT-GROUP
OBJECTS { mcastVpnMdtJnSendMdtGrpAddrType,
          mcastVpnMdtJnSendMdtGroup,
          mcastVpnMdtJnSendMdtRefCt
        }
STATUS   current
DESCRIPTION
    "Collection of objects needed for management of MVPN MDT
    data joins received by a device."
 ::= { mcastVpnGroups 8 }

mcastVpnMIBTunnelGroup OBJECT-GROUP
OBJECTS { mcastVpnTunnelName,
```



```
        mcastVpnTunnelMvrf
    }
    STATUS current
    DESCRIPTION
        "Objects required for MDT tunnel interface management."
    ::= { mcastVpnGroups 9 }

mcastVpnMIBNotificationGroup NOTIFICATION-GROUP
    NOTIFICATIONS { mcastVpnMvrfChange
    }
    STATUS current
    DESCRIPTION
        "A collection of MVPN notifications."
    ::= { mcastVpnGroups 10 }
```

END

7. Security Considerations

The management objects defined in this MIB module with a MAX-ACCESS clause of read-create may be considered sensitive or vulnerable in some network environments. These are such tables: mcastVpnGenericTable, mcastVpnMdtDefaultTable, mcastVpnMdtDataTable. Unauthorized access to these tables could result in disruption of MVPN services in the network.

The use of stronger mechanisms such as SNMPv3 security should be considered where possible. Administrators should consider whether read access to these objects should be allowed, since read access may be undesirable under certain circumstances.

Some of the readable objects in this MIB module "i.e., objects with a MAX-ACCESS other than not-accessible" may be considered sensitive or vulnerable in some network environments. It is thus important to control even GET and/or NOTIFY access to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP.

It is RECOMMENDED that implementers consider the security features as provided by the SNMPv3 framework including full support for the SNMPv3 cryptographic mechanisms for authentication and privacy.

8. IANA Considerations

This documentation has no actions for IANA at this point.

9. Acknowledgments

We wish to thank Toerless Eckert and Yiqun Cai for their suggestions.

10. References

10.1. Normative References

- [MCAST-VPN] Y. Cai, E. Rosen, I. Wijnands, "Multicast in BGP/MPLS IP VPNs", <[draft-rosen-vpn-mcast-07.txt](#)>, May 2004.
- [MDT-SAFI] G. Nalawade, A. Sreekantiah, "MDT SAFI" <[draft-nalawade-mdt-safi-00.txt](#)>, February 2004.
- [MT-DISC] I. Wijnands, G. Nalawade, "MT Tunnel Discovery and RPF check", <[draft-wijnands-mt-discovery-00.txt](#)>, August 2004.
- [RFC2547bis] E. Rosen, et. al., "BGP/MPLS VPNs", <[draft-ietf-l3vpn-rfc2547bis-01.txt](#)>, September 2003.
- [MPLS-L3VPN-MIB] T. Nadeau, H. Van Der Linde, "MPLS/BGP Layer 3 VPN MIB", <[draft-ietf-l3vpn-mpls-vpn-mib-04.txt](#)>, May 2004
- [RFC2119] S. Bradner, "Key Words for use in RFCs to Indicate Requirement Levels", [RFC 2119](#), [BCP 14](#), March 1997.
- [RFC2578] K. McCloghrie, D. Perkins, J. Schoenwaelder, J. Case, M. Rose, S. Waldbusser, "Structure of Management Information Version 2 (SMIv2)", STD 58, [RFC 2578](#), April 1999.
- [RFC2579] K. McCloghrie, D. Perkins, J. Schoenwaelder, J. Case, M. Rose, and S. Waldbusser, "Textual Conventions for SMIv2", STD 58, [RFC 2579](#), April 1999.
- [RFC2580] K. McCloghrie, D. Perkins, J. Schoenwaelder, J. Case, M. Rose, and S. Waldbusser, "Conformance Statements for SMIv2", STD 58, [RFC 2580](#), April 1999.
- [RFC3291] M. Daniele, B. Haberman, S. Routhier, J. Schoenwaelder, "Textual Conventions for Internet Network Addresses", [RFC3291](#), May 2002.

- [RFC3411] D. Harrington, R. Presuhn, and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", [RFC 3411](#), December 2002.
- [TCMIB] T. Nadeau, J. Cucchiara, C. Srinivasan, A. Viswanathan, H. Sjostrand, H. and K. Kompella, "Definition of Textual Conventions and OBJECT-IDENTITIES for Multi-Protocol Label Switching (MPLS) Management", Internet Draft <[draft-ietf-mpls-tc-mib-06.txt](#)>, April 2003.

10.2. Informative References

- [RFC2667] D. Thaler, "IP-TUNNEL-MIB", [RFC 2667](#), August 1999.
- [RFC2863] K. McCloghrie and F. Kastenholtz, "The Interfaces Group MIB", [RFC 2863](#), June 2000.

11. Authors' Addresses

Susheela Vaidya
Cisco Systems, Inc.
170 W. Tasman Drive
San Jose, CA 95134
Phone: +1-408-525-1952
Email: svaidya@cisco.com

Thomas D. Nadeau
Cisco Systems, Inc.
300 Beaver Brook Road
Boxboro, MA 01719
Phone: +1-978-936-1470
Email: tnadeau@cisco.com

Harmen Van der Linde
AT&T - Layer-2/Layer-3 NM Architecture and Operations Planning
Room C2-3C34
200 Laurel Ave
Middletown, NJ 07748
Tel: +1-732-420-1916
Email: hvdl@att.com

12. Full Copyright Statement

Copyright (C) The Internet Society (2004). This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

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 AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM 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.

13. 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](#) [[RFC2028](#)]. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

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