Softwire Mesh Management Information Base (MIB)

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 defines objects for managing a softwire mesh.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7856.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.
1. Introduction

The softwire mesh framework [RFC5565] is a tunneling mechanism that enables connectivity between islands of IPv4 networks across a single IPv6 backbone and vice versa. In a softwire mesh, extended Multiprotocol BGP (MP-BGP) is used to set up tunnels and advertise prefixes among Address Family Border Routers (AFBRs).

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it defines objects for managing a softwire mesh [RFC5565].

2. The Internet-Standard 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. Terminology

This document uses terminology from the softwire problem statement [RFC4925], the BGP encapsulation Subsequent Address Family Identifier (SAFI), the BGP tunnel encapsulation attribute [RFC5512], the softwire mesh framework [RFC5565], and the BGP IPsec tunnel encapsulation attribute [RFC5566].

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

4. Structure of the MIB Module

The Softwire Mesh MIB provides a method to monitor the softwire mesh objects through SNMP.

4.1. The swmSupportedTunnelTable Subtree

The swmSupportedTunnelTable subtree provides the information about what types of tunnels can be used for softwire mesh scenarios in the AFBR. The softwire mesh framework [RFC5565] does not mandate the use of any particular tunneling technology. Based on the BGP tunnel encapsulation attribute tunnel types introduced by RFC 5512 [RFC5512] and RFC 5566 [RFC5566], the softwire mesh tunnel types include at least L2TPv3 (Layer 2 Tunneling Protocol version 3) over IP, GRE (Generic Routing Encapsulation), Transmit tunnel endpoint, IPsec in Tunnel-mode, IP in IP tunnel with IPsec Transport Mode, MPLS-in-IP tunnel with IPsec Transport Mode, and IP in IP. The detailed encapsulation information of different tunnel types (e.g., L2TPv3 Session ID, GRE Key, etc.) is not managed in the Softwire Mesh MIB.

4.2. The swmEncapsTable Subtree

The swmEncapsTable subtree provides softwire mesh NLRI-NH information (Network Layer Reachability Information - Next Hop) about the AFBR. It keeps the mapping between the External-IP (E-IP) prefix and the Internal-IP (I-IP) address of the next hop. The mappings determine which I-IP destination address will be used to encapsulate the received packet according to its E-IP destination address. The definitions of E-IP and I-IP are explained in Section 4.1 of RFC 5565 [RFC5565]. The number of entries in swmEncapsTable shows how many softwire mesh tunnels are maintained in this AFBR.
4.3. The swmBGPNeighborTable Subtree

This subtree provides the softwire mesh BGP neighbor information of an AFBR. It includes the address of the softwire mesh BGP peer and the kind of tunnel that the AFBR would use to communicate with this BGP peer.

4.4. The swmConformance Subtree

This subtree provides the conformance information of MIB objects.

5. Relationship to Other MIB Modules

5.1. Relationship to the IF-MIB

The Interfaces MIB [RFC2863] defines generic managed objects for managing interfaces. Each logical interface (physical or virtual) has an ifEntry. Tunnels are handled by creating logical interfaces (ifEntry). Being a tunnel, the softwire mesh interface has an entry in the Interface MIB, as well as an entry in the IP Tunnel MIB. Those corresponding entries are indexed by ifIndex.

The ifOperStatus in the ifTable represents whether the mesh function of the AFBR has been triggered. If the softwire mesh capability is negotiated during the BGP OPEN phase, the mesh function is considered to be started, and the ifOperStatus is "up". Otherwise, the ifOperStatus is "down".

In the case of an IPv4-over-IPv6 softwire mesh tunnel, ifInUcastPkts counts the number of IPv6 packets that are sent to the virtual interface for decapsulation into IPv4. The ifOutUcastPkts counts the number of IPv6 packets that are generated by encapsulating IPv4 packets sent to the virtual interface. In particular, if these IPv4 packets need fragmentation, ifOutUcastPkts counts the number of packets after fragmentation.

In the case of an IPv6-over-IPv4 softwire mesh tunnel, ifInUcastPkts counts the number of IPv4 packets that are delivered to the virtual interface for decapsulation into IPv6. The ifOutUcastPkts counts the number of IPv4 packets that are generated by encapsulating IPv6 packets sent down to the virtual interface. In particular, if these IPv6 packets need to be fragmented, ifOutUcastPkts counts the number of packets after fragmentation. Similar definitions apply to other counter objects in the ifTable.
5.2. Relationship to the IP Tunnel MIB

The IP Tunnel MIB [RFC4087] contains objects applicable to all IP tunnels, including softwire mesh tunnels. Meanwhile, the Softwire Mesh MIB extends the IP Tunnel MIB to further describe encapsulation-specific information.

When running a point-to-multipoint tunnel, it is necessary for a softwire mesh AFBR to maintain an encapsulation table in order to perform correct "forwarding" among AFBRs. This forwarding function on an AFBR is performed by using the E-IP destination address to look up the I-IP encapsulation destination address in the encapsulation table. An AFBR also needs to know the BGP peer information of the other AFBRs, so that it can negotiate the NLRI-NH information and the tunnel parameters with them.

The Softwire Mesh MIB requires the implementation of the IP Tunnel MIB. The tunnelIfEncapsMethod in the tunnelIfEntry MUST be set to softwireMesh(16), and a corresponding entry in the Softwire Mesh MIB module will be presented for the tunnelIfEntry. The tunnelIfRemoteInetAddress MUST be set to "0.0.0.0" for IPv4 or "::" for IPv6 because it is a point-to-multipoint tunnel.

The tunnelIfAddressType in the tunnelIfTable represents the type of address in the corresponding tunnelIfLocalInetAddress and tunnelIfRemoteInetAddress objects. The tunnelIfAddressType is identical to swmEncapsIIPDstType in softwire mesh, which can support either IPv4-over-IPv6 or IPv6-over-IPv4. When the swmEncapsEIPDstType is IPv6 and the swmEncapsIIPDstType is IPv4, the tunnel type is IPv6-over-IPv4; when the swmEncapsEIPDstType is IPv4 and the swmEncapsIIPDstType is IPv6, the encapsulation mode is IPv4-over-IPv6.

5.3. MIB Modules Required for IMPORTS

The following MIB module IMPORTS objects from SNMPv2-SMI [RFC2578], SNMPv2-CONF [RFC2580], IF-MIB [RFC2863], and INET-ADDRESS-MIB [RFC4001].
6. Definitions

SOFTWIRE-MESH-MIB DEFINITIONS ::= BEGIN

IMPORTS
   MODULE-IDENTITY, OBJECT-TYPE, mib-2 FROM SNMPv2-SMI
   OBJECT-GROUP, MODULE-COMPLIANCE             FROM SNMPv2-CONF
   InetAddress, InetAddressType, InetAddressPrefixLength
   FROM INET-ADDRESS-MIB
   ifIndex                                FROM IF-MIB
   IANA_tunnelType                         FROM IANAifType-MIB;

swmMIB MODULE-IDENTITY
LAST-UPDATED "201605110000Z" -- May 11, 2016
ORGANIZATION "Softwire Working Group"
CONTACT-INFO
 "Yong Cui
   Email: yong@csnet1.cs.tsinghua.edu.cn

   Jiang Dong
   Email: knight.dongjiang@gmail.com

   Peng Wu
   Email: weapon9@gmail.com

   Mingwei Xu
   Email: xmw@cernet.edu.cn

   Antti Yla-Jaaski
   Email: antti.yla-jaaski@aalto.fi

   Email comments directly to the Softwire WG Mailing
   List at softwires@ietf.org"

DESCRIPTION
 "This MIB module contains managed object definitions for
the softwire mesh framework.

Copyright (c) 2016 IETF Trust and the persons
identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with
or without modification, is permitted pursuant to, and
subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info).

REVISION "2016051110000Z"

DESCRIPTION "Initial version, published as RFC 7856"

::= { mib-2 239 }

swmObjects OBJECT IDENTIFIER ::= { swmMIB 1 }

-- swmSupportedTunnelTable

swmSupportedTunnelTable OBJECT-TYPE
SYNTAX SEQUENCE OF SwmSupportedTunnelEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "A table of objects that show what kinds of tunnels can be supported by the AFBR."
::= { swmObjects 1 }

swmSupportedTunnelEntry OBJECT-TYPE
SYNTAX SwmSupportedTunnelEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "A set of objects that show what kinds of tunnels can be supported in the AFBR. If the AFBR supports multiple tunnel types, the swmSupportedTunnelTable would have several entries."
INDEX { swmSupportedTunnelType }
::= { swmSupportedTunnelTable 1 }

SwmSupportedTunnelEntry ::=SEQUENCE {
    swmSupportedTunnelType IANAtunnelType
}

swmSupportedTunnelType OBJECT-TYPE
SYNTAX IANAtunnelType
MAX-ACCESS read-only
STATUS current
DESCRIPTION "Represents the tunnel type that can be used for softwire mesh scenarios, such as L2TPv3 over IP, GRE, Transit tunnel endpoint, IPsec in Tunnel-mode, IP in IP tunnel with IPsec Transport Mode, MPLS-in-IP tunnel with IPsec Transport Mode, and IP in IP. There is no restriction on the tunnel type the softwire mesh can use."

REFERENCE
"L2TPv3 over IP, GRE, and IP in IP in RFC 5512. Transmit tunnel endpoint, IPsec in Tunnel-mode, IP in IP tunnel with IPsec Transport Mode, MPLS-in-IP tunnel with IPsec Transport Mode in RFC 5566."
::= { swmSupportedTunnelEntry 1 }

-- end of swmSupportedTunnelTable

--swmEncapsTable
swmEncapsTable OBJECT-TYPE
SYNTAX      SEQUENCE OF SwmEncapsEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
"A table of objects that display the softwire mesh encapsulation information."
::= { swmObjects 2 }

swmEncapsEntry OBJECT-TYPE
SYNTAX      SwmEncapsEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
"A table of objects that manage the softwire mesh I-IP encapsulation destination based on the E-IP destination prefix."
INDEX { ifIndex,
          swmEncapsEIPDstType,
          swmEncapsEIPDst,
          swmEncapsEIPPrefixLength
         }
::= { swmEncapsTable 1 }

SwmEncapsEntry ::= SEQUENCE {
    swmEncapsEIPDstType InetAddressType,
    swmEncapsEIPDst InetAddress,
    swmEncapsEIPPrefixLength InetAddressPrefixLength,
    swmEncapsIIPDstType InetAddressType,
    swmEncapsIIPDst InetAddress
}

swmEncapsEIPDstType OBJECT-TYPE
SYNTAX      InetAddressType
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
"This object specifies the address type used for swmEncapsEIPDst. It is different from the
tunnelIfAddressType in the tunnelIfTable. The
swmEncapsEIPDstType is IPv6 (2) if it is IPv6-over-IPv4
tunneling. The swmEncapsEIPDstType is
IPv4 (1) if it is IPv4-over-IPv6 tunneling."
REFERENCE
"IPv4 and IPv6 in RFC 4001."
::= { swmEncapsEntry 1 }

swmEncapsEIPDst OBJECT-TYPE
SYNTAX    InetAddress
MAX-ACCESS not-accessible
STATUS    current
DESCRIPTION
"The E-IP destination prefix, which is
used for I-IP encapsulation destination looking up.
The type of this address is determined by the
value of swmEncapsEIPDstType"
REFERENCE
"E-IP and I-IP in RFC 5565."
::= { swmEncapsEntry 2 }

swmEncapsEIPPrefixLength OBJECT-TYPE
SYNTAX    InetAddressPrefixLength
MAX-ACCESS not-accessible
STATUS    current
DESCRIPTION
"The prefix length of the E-IP destination prefix."
::= { swmEncapsEntry 3 }

swmEncapsIIPDstType OBJECT-TYPE
SYNTAX    InetAddressType
MAX-ACCESS read-only
STATUS    current
DESCRIPTION
"This object specifies the address type used for
swmEncapsIIPDst. It is the same as the tunnelIfAddressType
in the tunnelIfTable."
REFERENCE
"IPv4 and IPv6 in RFC 4001."
::= { swmEncapsEntry 4 }

swmEncapsIIPDst OBJECT-TYPE
SYNTAX    InetAddress
MAX-ACCESS read-only
STATUS    current
DESCRIPTION
"The I-IP destination address, which is used as the
encapsulation destination for the corresponding E-IP
prefix. Since the tunnelIfRemoteInetAddress in the tunnelIfTable should be 0.0.0.0 or ::, swmEncapIIPDst should be the destination address used in the outer IP header."

REFERENCE
"E-IP and I-IP in RFC 5565."

::= { swmEncapsEntry 5 }

-- End of swmEncapsTable

-- swmBGPNeighborTable

swmBGPNeighborTable OBJECT-TYPE
SYNTAX      SEQUENCE OF SwmBGPNeighborEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
"A table of objects that display the softwire mesh BGP neighbor information."
::= { swmObjects 3 }

swmBGPNeighborEntry OBJECT-TYPE
SYNTAX      SwmBGPNeighborEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
"A set of objects that display the softwire mesh BGP neighbor information."
INDEX {
  ifIndex,
  swmBGPNeighborInetAddressType,
  swmBGPNeighborInetAddress
}
::= { swmBGPNeighborTable 1 }

SwmBGPNeighborEntry ::= SEQUENCE {
  swmBGPNeighborInetAddressType  InetAddressType,
  swmBGPNeighborInetAddress      InetAddress,
  swmBGPNeighborTunnelType        IANAtunnelType
}

swmBGPNeighborInetAddressType OBJECT-TYPE
SYNTAX      InetAddressType
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
"This object specifies the address type used for swmBGPNeighborInetAddress."
::= { swmBGPNeighborEntry 1 }
swmBGPNeighborInetAddress OBJECT-TYPE
   SYNTAX     InetAddress
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
               "The address of the AFBR's BGP neighbor. The
               address type is the same as the tunnelIfAddressType
               in the tunnelIfTable."
               ::= { swmBGPNeighborEntry 2 }

swmBGPNeighborTunnelType OBJECT-TYPE
   SYNTAX     IANAtunnelType
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
               "Represents the type of tunnel that the AFBR
               chooses to transmit traffic with another AFBR/BGP
               neighbor."
               ::= { swmBGPNeighborEntry 3 }
-- End of swmBGPNeighborTable

-- conformance information
swmConformance
   OBJECT IDENTIFIER ::= { swmMIB 2 }

swmCompliances
   OBJECT IDENTIFIER ::= { swmConformance 1 }

swmGroups
   OBJECT IDENTIFIER ::= { swmConformance 2 }

-- compliance statements
swmCompliance MODULE-COMPLIANCE
   STATUS current
   DESCRIPTION
               "Describes the requirements for conformance to the Softwire
               Mesh MIB."

   The following index objects cannot be added as OBJECT
   clauses but nevertheless have compliance requirements:
   "
   -- OBJECT  swmEncapsEIPDstType
   -- SYNTAX  InetAddressType { ipv4(1), ipv6(2) }
   -- DESCRIPTION
   -- "An implementation is required to support
   -- global IPv4 and/or IPv6 addresses, depending
   -- on its support for IPv4 and IPv6."
   -- OBJECT  swmEncapsEIPDst
-- SYNTAX  InetAddress (SIZE(4|16))
-- DESCRIPTION
-- "An implementation is required to support
global IPv4 and/or IPv6 addresses, depending
on its support for IPv4 and IPv6."

-- OBJECT  swmEncapsEIPPrefixLength
-- SYNTAX  InetAddressPrefixLength (Unsigned32 (0..128))
-- DESCRIPTION
-- "An implementation is required to support
global IPv4 and/or IPv6 addresses, depending
on its support for IPv4 and IPv6."

-- OBJECT  swmBGPNeighborInetAddressType
-- SYNTAX  InetAddressType { ipv4(1), ipv6(2) }
-- DESCRIPTION
-- "An implementation is required to support
global IPv4 and/or IPv6 addresses, depending
on its support for IPv4 and IPv6."

-- OBJECT  swmBGPNeighborInetAddress
-- SYNTAX  InetAddress (SIZE(4|16))
-- DESCRIPTION
-- "An implementation is required to support
global IPv4 and/or IPv6 addresses, depending
on its support for IPv4 and IPv6."

MODULE -- this module
MANDATORY-GROUPS    {
    swmSupportedTunnelGroup,
    swmEncapsGroup,
    swmBGPNeighborGroup
}
 ::= { swmCompliances 1 }

swmSupportedTunnelGroup  OBJECT-GROUP
OBJECTS {
    swmSupportedTunnelType
}
STATUS  current
DESCRIPTION
    "The collection of objects that are used to show
what kind of tunnel the AFBR supports."
 ::= { swmGroups 1 }

swmEncapsGroup  OBJECT-GROUP
OBJECTS {
    swmEncapsEIPDst,
7. Security Considerations

Because this MIB module reuses the IP Tunnel MIB, the security considerations of the IP Tunnel MIB are also applicable to the Softwire Mesh MIB.

There are no management objects defined in this MIB module that have a MAX-ACCESS clause of read-write and/or read-create. So, if this MIB module is implemented correctly, then there is no risk that an intruder can alter or create any management objects of this MIB module via direct SNMP SET operations.

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. These are the objects and their sensitivity/vulnerability:

swmSupportedTunnelType, swmEncapsIIPDstType, swmEncapsIIPDst, and swmBGPNeighborTunnelType can expose the types of tunnels used within the internal network and potentially reveal the topology of the internal network.
SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPsec), there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

Implementations SHOULD provide the security features described by the SNMPv3 framework (see [RFC3410]), and implementations claiming compliance to the SNMPv3 standard MUST include full support for authentication and privacy via the User-based Security Model (USM) [RFC3414] with the AES cipher algorithm [RFC3826]. Implementations MAY also provide support for the Transport Security Model (TSM) [RFC5591] in combination with a secure transport such as SSH [RFC5592] or TLS/DTLS [RFC6353].

Further, deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

8. IANA Considerations

IANA has allocated the following OBJECT IDENTIFIER value and recorded it in the SMI Numbers registry in the subregistry called "SMI Network Management MGMT Codes Internet-standard MIB" under the mib-2 branch (1.3.6.1.2.1):

<table>
<thead>
<tr>
<th>Descriptor</th>
<th>OBJECT IDENTIFIER value</th>
</tr>
</thead>
<tbody>
<tr>
<td>swmMIB</td>
<td>{ mib-2 239 }</td>
</tr>
</tbody>
</table>

IANA has recorded the following IANAInterfaceType Textual Convention within the IANAIfType-MIB:

```
IANAInterfaceType ::= TEXTUAL-CONVENTION
SYNTAX      INTEGER {
  softwireMesh(16)  -- softwire mesh tunnel
}
```
9. References

9.1. Normative References


9.2. Informative References


Acknowledgements

The authors would like to thank Dave Thaler, Jean-Philippe Dionne, Qi Sun, Sheng Jiang, and Yu Fu for their valuable comments.

Authors’ Addresses

Yong Cui
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing  100084
China

Phone: +86-10-6260-3059
Email: yong@csnet1.cs.tsinghua.edu.cn

Jiang Dong
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing  100084
China

Phone: +86-10-6278-5822
Email: knight.dongjiang@gmail.com

Peng Wu
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing  100084
China

Phone: +86-10-6278-5822
Email: weapon9@gmail.com

Mingwei Xu
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing  100084
China

Phone: +86-10-6278-5822
Email: xmw@cernet.edu.cn