Dual-Stack Lite (DS-Lite) Management Information Base (MIB) for Address Family Transition Routers (AFTRs)

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 managed objects for Address Family Transition Routers (AFTRs) of Dual-Stack Lite (DS-Lite).

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 7841.

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/rfc7870.

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

Dual-Stack Lite [RFC6333] is a solution that offers both IPv4 and IPv6 connectivity to customers crossing an IPv6-only infrastructure. One of its key components is an IPv4-over-IPv6 tunnel, which is used to provide IPv4 connectivity across a service provider’s IPv6 network. Another key component is a carrier-grade IPv4-IPv4 Network Address Translation (NAT) to share service provider IPv4 addresses among customers.

This document defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. This MIB module may be used for configuration and monitoring of Address Family Transition Routers (AFTRs) in a Dual-Stack Lite scenario.

2. Requirements Language

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 BCP 14, RFC 2119 [RFC2119]. When these words are not in ALL CAPS (such as "should" or "Should"), they have their usual English meanings and are not to be interpreted as [RFC2119] key words.
3. 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 [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].

4. 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 a logical interface (ifEntry) for each tunnel. Each DS-Lite tunnel endpoint also acts as a virtual interface that has a corresponding entry in the IP Tunnel MIB and Interface MIB. Those corresponding entries are indexed by ifIndex.

The ifOperStatus in ifTable is used to represent whether the DS-Lite tunnel function has been triggered. The ifInUcastPkts defined in ifTable will represent the number of IPv4 packets that have been encapsulated into IPv6 packets sent to a Basic Bridging BroadBand (B4). The ifOutUcastPkts defined in ifTable contains the number of IPv6 packets that can be decapsulated to IPv4 in the virtual interface. Also, the IF-MIB defines ifMtu for the MTU of this tunnel interface, so the DS-Lite MIB does not need to define the MTU for the tunnel.

5. Difference from the IP Tunnel MIB and NATv2-MIB

The key technologies for DS-Lite are IP-in-IP (IPv4-in-IPv6) tunnels and NAT (IPv4-to-IPv4 translation).

Notes: According to Section 5.2 of [RFC6333], DS-Lite only defines IPv4 in IPv6 tunnels at this moment, but other types of encapsulation could be defined in the future. So, the DS-Lite MIB only supports IP-in-IP encapsulation. If another RFC defines other tunnel types in the future, the DS-Lite MIB will be updated then.
The NATV2-MIB [RFC7659] is designed to carry translation from any address family to any address family; therefore, it supports IPv4-to-IPv4 translation.

The IP Tunnel MIB [RFC4087] is designed to manage tunnels of any type over IPv4 and IPv6 networks; therefore, it already supports IP-in-IP tunnels. But in a DS-Lite scenario, the tunnel type is point-to-multipoint IP-in-IP tunnels. The direct(2) defined in the IP Tunnel MIB only supports point-to-point tunnels. So, it needs to define a new tunnel type for DS-Lite.

However, the NATV2-MIB and IP Tunnel MIB together are not sufficient to support DS-Lite. This document describes the specific features for the DS-Lite MIB, as below.

In the DS-Lite scenario, the Address Family Transition Router (AFTR) is not only the tunnel-end concentrator, but also an IPv4-to-IPv4 NAT. So, as defined in [RFC6333], when the IPv4 packets come back from the Internet to the AFTR, it knows how to reconstruct the IPv6 encapsulation by doing a reverse lookup in the extended IPv4 NAT binding table (Section 6.6 of [RFC6333]). The NAT binding table in the AFTR is extended to include the IPv6 address of the tunnel initiator. However, the NAT binding information defined in the NATV2-MIB as natv2PortMapTable is indexed by the NAT instance, protocol, and external realm and address. Because the tunnelIfTable defined in the TUNNEL-MIB [RFC4087] is indexed by the ifIndex, the DS-Lite MIB needs to define the tunnel objects to extend the NAT binding entry by interface. Therefore, a combined MIB is necessary.

An implementation of the IP Tunnel MIB is required for DS-Lite. As the tunnel is not point-to-point in DS-Lite, it needs to define a new tunnel type for DS-Lite. The tunnelIfEncapsMethod in the tunnelIfEntry should be set to dsLite(17), and a corresponding entry in the DS-Lite module will exist for every tunnelIfEntry with this tunnelIfEncapsMethod. The tunnelIfRemoteInetAddress must be set to "::".

6. Structure of the MIB Module

The DS-Lite MIB provides a way to monitor and manage the devices (AFTRs) in a DS-Lite scenario through SNMP.

The DS-Lite MIB is configurable on a per-interface basis. It depends on several parts of the IF-MIB [RFC2863], IP Tunnel MIB [RFC4087], and NATV2-MIB [RFC7659].
6.1. The Object Group

This group defines objects that are needed for the DS-Lite MIB.

6.1.1. The dsliteTunnel Subtree

The dsliteTunnel subtree describes managed objects used for managing tunnels in the DS-Lite scenario. Because the tunnelInetConfigLocalAddress and the tunnelInetConfigRemoteAddress defined in the IP Tunnel MIB are not readable, a few new objects are defined in the DS-Lite MIB.

6.1.2. The dsliteNAT Subtree

The dsliteNAT subtree describes managed objects used for configuration and monitoring of an AFTR that is capable of a NAT function. Because the NATv2-MIB supports the NAT management function in DS-Lite, we may reuse it in the DS-Lite MIB. The dsliteNAT subtree also provides the mapping information between the tunnel entry (dsliteTunnelEntry) and the NAT entry (dsliteNATBindEntry) by adding the IPv6 address of the B4 to the natv2PortMapEntry in the NATv2-MIB. The mapping behavior, filtering behavior, and pooling behavior described in this subtree are all defined in [RFC4787].

6.1.3. The dsliteInfo Subtree

The dsliteInfo subtree provides statistical information for DS-Lite.

6.2. The Notification Group

This group defines some notification objects for a DS-Lite scenario.

6.3. The Conformance Group

The dsliteConformance subtree provides conformance information of MIB objects.

7. MIB Modules Required for IMPORTS

This MIB module IMPORTs objects from [RFC2578], [RFC2580], [RFC2863], [RFC3411], [RFC4001], and [RFC7659].
8. Definitions

DSLite-MIB DEFINITIONS ::= BEGIN

IMPORTS
   MODULE-IDENTITY, OBJECT-TYPE, mib-2,
   NOTIFICATION-TYPE, Integer32,
   Counter64, Unsigned32
   FROM SNMPv2-SMI

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

   SnmpAdminString
   FROM SNMP-FRAMEWORK-MIB

   ifIndex
   FROM IF-MIB

   InetAddress, InetAddressType, InetAddressPrefixLength,
   InetPortNumber
   FROM INET-ADDRESS-MIB

   ProtocolNumber, Natv2InstanceIndex, Natv2SubscriberIndex
   FROM NATV2-MIB;

dsliteMIB MODULE-IDENTITY
LAST-UPDATED "201605110000Z"          -- May 11, 2016
ORGANIZATION "IETF Softwire Working Group"
CONTACT-INFO
"Yu Fu
CNNIC
No.4 South 4th Street, Zhongguancun
Hai-Dian District, Beijing 100090
China
Email: fuyu@cnnic.cn

Sheng Jiang
Huawei Technologies Co., Ltd
Huawei Building, 156 Beiqing Rd.
Hai-Dian District, Beijing 100095
China
Email: jiangsheng@huawei.com
"The MIB module is defined for management of objects in the DS-Lite scenario.

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 "201605110000Z"

DESCRIPTION
"Initial version. Published as RFC 7870."

::= { mib-2 240 }

--Top-level components of this MIB module

dsliteMIBObjects OBJECT IDENTIFIER ::= { dsliteMIB 1 }
dsliteTunnel OBJECT IDENTIFIER ::= { dsliteMIBObjects 1 }
dsliteNAT OBJECT IDENTIFIER ::= { dsliteMIBObjects 2 }
dsliteInfo OBJECT IDENTIFIER ::= { dsliteMIBObjects 3 }

--Notifications section
ds-liteNotifications  OBJECT IDENTIFIER  ::=  { ds-liteMIB 0 }  

--ds-liteTunnel

--ds-liteTunnelTable

ds-liteTunnelTable  OBJECT-TYPE
  SYNTAX  SEQUENCE OF DsliteTunnelEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION "The (conceptual) table containing information on configured tunnels. This table can be used to map a B4 address to the associated AFTR address. It can also be used for row creation."
  REFERENCE "B4, AFTR: RFC 6333."
  ::=  { ds-liteTunnelTable 1 }

DsliteTunnelEntry  OBJECT-TYPE
  SYNTAX  DsliteTunnelEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION "Each entry in this table contains the information on a particular configured tunnel."
  INDEX  { ds-liteTunnelAddressType, ds-liteTunnelStartAddress, ds-liteTunnelEndAddress, ifIndex }
  ::=  { ds-liteTunnelTable 1 }

DsliteTunnelEntry  ::=  
  SEQUENCE  {
    ds-liteTunnelAddressType     InetAddressType,
    ds-liteTunnelStartAddress    InetAddress,
    ds-liteTunnelEndAddress      InetAddress,
    ds-liteTunnelStartAddPreLen  InetAddressPrefixLength
  }

ds-liteTunnelAddressType  OBJECT-TYPE
  SYNTAX  InetAddressType
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION "This object MUST be set to the value of ipv6(2)."
It describes the address type of the IPv4-in-IPv6
tunnel initiator and endpoint.

REFERENCE
"ipv6(2): RFC 4001."
::= { dsliteTunnelEntry 1 }

dsliteTunnelStartAddress OBJECT-TYPE
SYNTAX InetAddress (SIZE (0..16))
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"The IPv6 address of the initiator of the tunnel.
The address type is given by dsliteTunnelAddressType."
::= { dsliteTunnelEntry 2 }

dsliteTunnelEndAddress OBJECT-TYPE
SYNTAX InetAddress (SIZE (0..16))
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"The IPv6 address of the endpoint of the tunnel.
The address type is given by dsliteTunnelAddressType."
::= { dsliteTunnelEntry 3 }

dsliteTunnelStartAddPreLen OBJECT-TYPE
SYNTAX InetAddressPrefixLength
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The IPv6 prefix length of the IP address for the initiator of the tunnel(dsliteTunnelStartAddress)."
::= { dsliteTunnelEntry 4 }

--dsliteNATBindTable(according to the NAPT scheme)

dsliteNATBindTable OBJECT-TYPE
SYNTAX SEQUENCE OF DsliteNATBindEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"This table contains information about currently active NAT binds in the NAT of the AFTR. This table adds the IPv6 address of a B4 to the natv2PortMapTable defined in NATV2-MIB (RFC 7659)."
REFERENCE
"NATV2-MIB: Section 4 of RFC 7659."
::= ( dsliteNAT 1 )
ds-liteNATBindEntry OBJECT-TYPE
SYNTAX DsliteNATBindEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"The entry in this table holds the mapping relationship
between tunnel information and NAT bind information.
Each entry in this table not only needs to match a
corresponding entry in the natv2PortMapTable, but
also a corresponding entry in the dsliteTunnelTable.
So, the INDEX of the entry needs to match a corresponding
value in the natv2PortMapTable INDEX and a corresponding
value in the dsliteTunnelTable INDEX. These entries are
lost upon agent restart."
REFERENCE
"natv2PortMapTable: Section 4 of RFC 7659."
INDEX { dsliteNATBindMappingInstanceIndex,
dsliteNATBindMappingProto,
dsliteNATBindMappingExtRealm,
dsliteNATBindMappingExtAddressType,
dsliteNATBindMappingExtAddress,
dsliteNATBindMappingExtPort,
ifIndex,
dsliteTunnelStartAddress }
::= { dsliteNATBindTable 1 }

DsliteNATBindEntry ::= SEQUENCE {
    dsliteNATBindMappingInstanceIndex  Natv2InstanceIndex,
    dsliteNATBindMappingProto          ProtocolNumber,
    dsliteNATBindMappingExtRealm       SnmpAdminString,
    dsliteNATBindMappingExtAddressType InetAddressType,
    dsliteNATBindMappingExtAddress     InetAddress,
    dsliteNATBindMappingExtPort        InetPortNumber,
    dsliteNATBindMappingIntRealm       SnmpAdminString,
    dsliteNATBindMappingIntAddressType InetAddressType,
    dsliteNATBindMappingIntAddress     InetAddress,
    dsliteNATBindMappingIntPort        InetPortNumber,
    dsliteNATBindMappingPool           Unsigned32,
    dsliteNATBindMappingMapBehavior    INTEGER,
    dsliteNATBindMappingFilterBehavior INTEGER,
    dsliteNATBindMappingAddressPooling INTEGER
}

ds-liteNATBindMappingInstanceIndex OBJECT-TYPE
SYNTAX Natv2InstanceIndex
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"Index of the NAT instance that created this port map entry."
::= { dsliteNATBindEntry 1 }

dsliteNATBindMappingProto OBJECT-TYPE
SYNTAX ProtocolNumber
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"This object specifies the mapping’s transport protocol number."
::= { dsliteNATBindEntry 2 }

dsliteNATBindMappingExtRealm OBJECT-TYPE
SYNTAX SnmpAdminString (SIZE(0..32))
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"The realm to which dsliteNATBindMappingExtAddress belongs."
::= { dsliteNATBindEntry 3 }

dsliteNATBindMappingExtAddressType OBJECT-TYPE
SYNTAX InetAddressType
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"Address type for the mapping’s external address. This object MUST be set to the value of iPv4(1). The values of ipv6(2), ipv4z(3), and ipv6z(4) are not allowed."
REFERENCE
"ipv4(1), ipv6(2), ipv4z(3), and ipv6z(4): RFC 4001."
::= { dsliteNATBindEntry 4 }

dsliteNATBindMappingExtAddress OBJECT-TYPE
SYNTAX InetAddress (SIZE (0..4))
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"The mapping’s external address. This is the source address for translated outgoing packets. The address type is given by dsliteNATBindMappingExtAddressType."
::= { dsliteNATBindEntry 5 }

dsliteNATBindMappingExtPort OBJECT-TYPE
SYNTAX InetPortNumber
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"The port number for the mapping’s external address. This is the destination port for translated outgoing packets."
::= { dsliteNATBindEntry 6 }
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"The mapping’s assigned external port number.
This is the source port for translated outgoing packets. This MUST be a non-zero value."
::= { dsliteNATBindEntry 6 }

dsliteNATBindMappingIntRealm OBJECT-TYPE
SYNTAX SnmpAdminString (SIZE(0..32))
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The realm to which natMappingIntAddress belongs. This realm defines the IPv6 address space from which the tunnel source address is taken. The realm of the encapsulated IPv4 address is restricted in scope to the tunnel, so there is no point in identifying it separately."
::= { dsliteNATBindEntry 7 }

dsliteNATBindMappingIntAddressType OBJECT-TYPE
SYNTAX InetAddressType
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"Address type of the mapping’s internal address. This object MUST be set to the value of iPv4z(3). The values of ipv4(1), ipv6(2), and ipv6z(4) are not allowed."
REFERENCE
"ipv4(1), ipv6(2), iPv4z(3), and ipv6z(4): RFC 4001."
::= { dsliteNATBindEntry 8 }

dsliteNATBindMappingIntAddress OBJECT-TYPE
SYNTAX InetAddress
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The mapping’s internal address. It is the IPv6 tunnel source address. The address type is given by dsliteNATBindMappingIntAddressType."
::= { dsliteNATBindEntry 9 }

dsliteNATBindMappingIntPort OBJECT-TYPE
SYNTAX InetPortNumber
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The mapping’s internal port number. This MUST be a 
non-zero value."
::= { dsliteNATBindEntry 10 }

dsliteNATBindMappingPool OBJECT-TYPE
SYNTAX Unsigned32 (0..4294967295)
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"Index of the pool that contains this mapping’s external 
address and port. If zero, no pool is associated with 
this mapping."
::= { dsliteNATBindEntry 11 }

dsliteNATBindMappingMapBehavior OBJECT-TYPE
SYNTAX INTEGER {
endpointIndependent (0),
addressDependent (1),
addressAndPortDependent (2)
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"Mapping behavior as described in Section 4.1 of RFC 4787.

epsilonendpointIndependent(0), the behavior REQUIRED by 
RFC 4787, REQ-1 maps the source address and port to 
the same external address and port for all destination 
address and port combinations reached through the same 
external realm and using the given protocol.

addressDependent(1) maps to the same external address 
and port for all destination ports at the same 
destination address reached through the same external 
realm and using the given protocol.

addressAndPortDependent(2) maps to a separate external 
address and port combination for each different 
destination address and port combination reached 
through the same external realm.

For the DS-Lite scenario, it must be 
addressAndPortDependent(2)."
REFERENCE
"Mapping behavior: Section 4.1 of RFC 4787.
DS-Lite: RFC 6333."
::= { dsliteNATBindEntry 12 }
ds-liteNATBindMappingFilterBehavior OBJECT-TYPE
SYNTAX INTEGER{
  endpointIndependent (0),
  addressDependent (1),
  addressAndPortDependent (2)
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"Filtering behavior as described in Section 5 of RFC 4787.

endpointIndependent(0) accepts for translation packets from all combinations of remote address and port destined to the mapped external address and port via the given external realm and using the given protocol.

addressDependent(1) accepts for translation packets from all remote ports from the same remote source address destined to the mapped external address and port via the given external realm and using the given protocol.

addressAndPortDependent(2) accepts for translation only those packets with the same remote source address, port, and protocol incoming from the same external realm as identified when the applicable port map entry was created.

RFC 4787, REQ-8 recommends either endpointIndependent(0) or addressDependent(1) filtering behavior, depending on whether application friendliness or security takes priority.

For the DS-Lite scenario, it must be addressAndPortDependent(2)."
REFERENCE
"Filtering behavior: Section 5 of RFC 4787.
DS-Lite: RFC 6333."
::= { dsliteNATBindEntry 13 }

ds-liteNATBindMappingAddressPooling OBJECT-TYPE
SYNTAX INTEGER{
  arbitrary (0),
  paired (1)
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"Type of address pooling behavior that was used to create this mapping.

arbitrary(0) pooling behavior means that the NAT instance may create the new port mapping using any address in the pool that has a free port for the protocol concerned.

paired(1) pooling behavior, the behavior RECOMMENDED by RFC 4787, REQ-2 means that once a given internal address has been mapped to a particular address in a particular pool, further mappings of the same internal address to that pool will reuse the previously assigned pool member address."

REFERENCE
"Pooling behavior: Section 4.1 of RFC 4787."

::= { dsliteNATBindEntry 14 }

--dsliteInfo

dsliteAFTRAAlarmScalar OBJECT IDENTIFIER ::= { dsliteInfo 1 }

dsliteAFTRAAlarmB4AddrType OBJECT-TYPE
  SYNTAX  InetAddressType
  MAX-ACCESS accessible-for-notify
  STATUS current
  DESCRIPTION
    "This object indicates the address type of the B4, which will send an alarm."
  ::= { dsliteAFTRAAlarmScalar 1 }

dsliteAFTRAAlarmB4Addr OBJECT-TYPE
  SYNTAX  InetAddress
  MAX-ACCESS accessible-for-notify
  STATUS current
  DESCRIPTION
    "This object indicates the IP address of B4, which will send an alarm. The address type is given by dsliteAFTRAAlarmB4AddrType."
  ::= { dsliteAFTRAAlarmScalar 2 }

dsliteAFTRAAlarmProtocolType OBJECT-TYPE
  SYNTAX  INTEGER{
    tcp (0),
    udp (1),
    icmp (2),
    total (3)
  }
MAX-ACCESS accessible-for-notify
STATUS current
DESCRIPTION
"This object indicates the transport protocol type of alarm.

tcp (0) means that the transport protocol type of alarm is tcp.

udp (1) means that the transport protocol type of alarm is udp.

icmp (2) means that the transport protocol type of alarm is icmp.

total (3) means that the transport protocol type of alarm is total."
::= { dsliteAFTRAlarmScalar 3 }

dsliteAFTRAlarmSpecificIPAddrType OBJECT-TYPE
SYNTAX InetAddressType
MAX-ACCESS accessible-for-notify
STATUS current
DESCRIPTION
"This object indicates the address type of the IP address whose port usage has reached the threshold."
::= { dsliteAFTRAlarmScalar 4 }

dsliteAFTRAlarmSpecificIP OBJECT-TYPE
SYNTAX InetAddress
MAX-ACCESS accessible-for-notify
STATUS current
DESCRIPTION
"This object indicates the IP address whose port usage has reached the threshold. The address type is given by dsliteAFTRAlarmSpecificIPAddrType."
::= { dsliteAFTRAlarmScalar 5 }

dsliteAFTRAlarmConnectNumber OBJECT-TYPE
SYNTAX Integer32 (60..90)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"This object indicates the notification threshold of the DS-Lite tunnels that is active in the AFTR device."
REFERENCE
"AFTR: Section 6 of RFC 6333."
DEFVAL
  { 60 }
::= { dsliteAFTRAAlarmScalar 6 }

dsliteAFTRAAlarmSessionNumber OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-write
STATUS current
DESCRIPTION
  "This object indicates the notification threshold of
  the IPv4 session for the user."
REFERENCE
  "AFTR: Section 6 of RFC 6333
  B4: Section 5 of RFC 6333."
DEFVAL
  { -1 }
::= { dsliteAFTRAAlarmScalar 7 }

dsliteAFTRAAlarmPortNumber OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-write
STATUS current
DESCRIPTION
  "This object indicates the notification threshold of the NAT
  ports that have been used by the user."
DEFVAL
  { -1 }
::= { dsliteAFTRAAlarmScalar 8 }

dsliteStatisticsTable OBJECT-TYPE
SYNTAX SEQUENCE OF DsliteStatisticsEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
  "This table provides statistical information
  about DS-Lite."
::= { dsliteInfo 2 }

dsliteStatisticsEntry OBJECT-TYPE
SYNTAX DsliteStatisticsEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
  "Each entry in this table provides statistical information
  about DS-Lite."
INDEX { dsliteStatisticsSubscriberIndex }
::= { dsliteStatisticsTable 1 }
DsliteStatisticsEntry ::=  
   SEQUENCE {
      dsliteStatisticsSubscriberIndex        Natv2SubscriberIndex,
      dsliteStatisticsDiscards               Counter64,
      dsliteStatisticsSends                  Counter64,
      dsliteStatisticsReceives               Counter64,
      dsliteStatisticsIpv4Session            Counter64,
      dsliteStatisticsIpv6Session            Counter64
   }

dsliteStatisticsSubscriberIndex OBJECT-TYPE
   SYNTAX Natv2SubscriberIndex
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Index of the subscriber or host. A unique value, greater than zero, for each subscriber in the managed system."
   ::= { dsliteStatisticsEntry 1 }

dsliteStatisticsDiscards OBJECT-TYPE
   SYNTAX Counter64
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This object indicates the number of packets discarded from this subscriber."
   ::= { dsliteStatisticsEntry 2 }

dsliteStatisticsSends OBJECT-TYPE
   SYNTAX Counter64
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This object indicates the number of packets that is sent to this subscriber."
   ::= { dsliteStatisticsEntry 3 }

dsliteStatisticsReceives OBJECT-TYPE
   SYNTAX Counter64
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This object indicates the number of packets that is received from this subscriber."
   ::= { dsliteStatisticsEntry 4 }
ds-liteStatisticsIpv4Session OBJECT-TYPE
SYNTAX  Counter64
MAX-ACCESS   read-only
STATUS   current
DESCRIPTION
  "This object indicates the number of the current IPv4 Sessions."
REFERENCE
  "Session: Paragraph 2 in Section 11 of RFC 6333.  
  (The AFTR should have the capability to log the 
tunnel-id, protocol, ports/IP addresses, and 
the creation time of the NAT binding to uniquely 
identify the user sessions)."
::= { dsliteStatisticsEntry 5 }

---

ds-liteStatisticsIpv6Session OBJECT-TYPE
SYNTAX  Counter64
MAX-ACCESS   read-only
STATUS   current
DESCRIPTION
  "This object indicates the number of the current IPv6 session.  
Because the AFTR is also a dual-stack device, it will also 
forward normal IPv6 packets for the inbound and outbound direction."
REFERENCE
  "Session: Paragraph 2 in Section 11 of RFC 6333.  
(The AFTR should have the capability to log the 
tunnel-id, protocol, ports/IP addresses, and 
the creation time of the NAT binding to uniquely 
identify the user sessions)."
::= { dsliteStatisticsEntry 6 }

---

ds-lite Notifications

ds-liteTunnelNumAlarm NOTIFICATION-TYPE
OBJECTS   { dsliteAFTRAlarmProtocolType,
            dsliteAFTRAlarmB4AddrType,
            dsliteAFTRAlarmB4Addr }
STATUS   current
DESCRIPTION
  "This trap is triggered when the number of current DS-Lite tunnels exceeds 
the value of the dsliteAFTRAlarmConnectNumber."
::= { dsliteNotifications 1 }
ds-liteAFTRUserSessionNumAlarm NOTIFICATION-TYPE
  OBJECTS { dsliteAFTRAlarmProtocolType,
    dsliteAFTRAlarmB4AddrType,
    dsliteAFTRAlarmB4Addr }  
  STATUS current  
  DESCRIPTION
    "This trap is triggered when user sessions
    reach the threshold. The threshold
    is specified by the dsliteAFTRAlarmSessionNumber."
  REFERENCE
    "Session: Paragraph 2 in Section 11 of RFC 6333.
    (The AFTR should have the capability to log the
    tunnel-id, protocol, ports/IP addresses, and
    the creation time of the NAT binding to uniquely
    identify the user sessions)."
  ::= { dsliteNotifications 2 }  

ds-liteAFTRPortUsageOfSpecificIpAlarm NOTIFICATION-TYPE
  OBJECTS { dsliteAFTRAlarmSpecificIPAddrType,
    dsliteAFTRAlarmSpecificIP }  
  STATUS current  
  DESCRIPTION
    "This trap is triggered when the used NAT
    ports of map address reach the threshold.
    The threshold is specified by the
    dsliteAFTRAlarmPortNumber."
  ::= { dsliteNotifications 3 }  

--Module Conformance statement  

dsliteConformance OBJECT IDENTIFIER
  ::= { dsliteMIB 2 }  

dsliteCompliances OBJECT IDENTIFIER
  ::= { dsliteConformance 1 }  

dsliteGroups OBJECT IDENTIFIER
  ::= { dsliteConformance 2 }  

-- compliance statements  

dsliteCompliance MODULE-COMPLIANCE
  STATUS current  
  DESCRIPTION
    "Describes the minimal requirements for conformance
    to the DS-Lite MIB."
  MODULE -- this module
    MANDATORY-GROUPS { dsliteNATBindGroup,
      dsliteTunnelGroup,
      dsliteStatisticsGroup,
      dsliteUserSessionGroup,
      dsliteUserSessionAlarmGroup,
      dslitePortUsageOfSpecificIpAlarmGroup,
      dsliteGroup }  

  ::= { dsliteCompliances 1 }  

Fu, et al. Standards Track [Page 20]
ds-liteNotificationsGroup,
dsl-liteAFTRAlarmScalarGroup } 
  ::= { dsl-liteCompliances 1 }

dsl-liteNATBindGroup OBJECT-GROUP
  OBJECTS {
    dsl-liteNATBindMappingIntRealm,
    dsl-liteNATBindMappingIntAddressType,
    dsl-liteNATBindMappingIntAddress,
    dsl-liteNATBindMappingIntPort,
    dsl-liteNATBindMappingPool,
    dsl-liteNATBindMappingMapBehavior,
    dsl-liteNATBindMappingFilterBehavior,
    dsl-liteNATBindMappingAddressPooling }
  STATUS current
  DESCRIPTION
    "A collection of objects to support basic
     management of NAT binds in the NAT of the AFTR."
  ::= { dsl-liteGroups 1 }

dsl-liteTunnelGroup OBJECT-GROUP
  OBJECTS { dsl-liteTunnelStartAddPreLen }
  STATUS current
  DESCRIPTION
    "A collection of objects to support management
     of DS-Lite tunnels."
  ::= { dsl-liteGroups 2 }

dsl-liteStatisticsGroup OBJECT-GROUP
  OBJECTS { dsl-liteStatisticsDiscards,
             dsl-liteStatisticsSends,
             dsl-liteStatisticsReceives,
             dsl-liteStatisticsIpv4Session,
             dsl-liteStatisticsIpv6Session }
  STATUS current
  DESCRIPTION
    "A collection of objects to support management
     of statistical information for AFTR devices."
  ::= { dsl-liteGroups 3 }

dsl-liteNotificationsGroup NOTIFICATION-GROUP
  NOTIFICATIONS { dsl-liteTunnelNumAlarm,
                  dsl-liteAFTRUserSessionNumAlarm,
                  dsl-liteAFTRPortUsageOfSpecificIpAlarm }
  STATUS current
  DESCRIPTION
    "A collection of objects to support management
     of trap information for AFTR devices."
::= { dsliteGroups 4 }

dsliteAFTRAlarmScalarGroup OBJECT-GROUP
  OBJECTS { dsliteAFTRAlarmB4AddrType,
            dsliteAFTRAlarmB4Addr,
            dsliteAFTRAlarmProtocolType,
            dsliteAFTRAlarmSpecificIPAddrType,
            dsliteAFTRAlarmSpecificIP,
            dsliteAFTRAlarmConnectNumber,
            dsliteAFTRAlarmSessionNumber,
            dsliteAFTRAlarmPortNumber }
  STATUS current
  DESCRIPTION
      "A collection of objects to support management of
       the information about the AFTR alarming scalar."
::= { dsliteGroups 5 }

END

9. Security Considerations

There are a number of management objects defined in this MIB module
with a MAX-ACCESS clause of read-write and/or read-create. Such
objects may be considered sensitive or vulnerable in some network
environments. The support for SET operations in a non-secure
environment without proper protection opens devices to attack. These
are the tables and objects and their sensitivity/vulnerability:

dsliteAFTRAlarmConnectNumber

dsliteAFTRAlarmSessionNumber

dsliteAFTRAlarmPortNumber

Notification thresholds: An attacker setting an arbitrarily low
threshold can cause many useless notifications to be generated.
Setting an arbitrarily high threshold can effectively disable
notifications, which could be used to hide another attack.

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 tables and objects and their
sensitivity/vulnerability:
entries in dsliteTunnelTable

entries in dsliteNATBindTable

Objects that reveal host identities: Various objects can reveal the identity of private hosts that are engaged in a session with external end nodes. A curious outsider could monitor these to assess the number of private hosts being supported by the AFTR device. Further, a disgruntled former employee of an enterprise could use the information to break into specific private hosts by intercepting the existing sessions or originating new sessions into the host. If nothing else, unauthorized monitoring of these objects will violate individual subscribers’ privacy.

Unauthorized read access to the dsliteTunnelTable would reveal information about the tunnel topology.

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.
10. 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>DSLite-MIB</td>
<td>{ mib-2 240 }</td>
</tr>
</tbody>
</table>

IANA has recorded the following IANA_tunnelType Textual Convention within the IANA_ifType-MIB:

```
IANA_tunnelType ::= TEXTUAL-CONVENTION
SYNTAX INTEGER {
    dsLite(17)        -- DS-Lite tunnel
}
```

11. References

11.1. Normative References


11.2. Informative References


Acknowledgements

The authors would like to thank the following for their valuable comments: Suresh Krishnan, Ian Farrer, Yiu Lee, Qi Sun, Yong Cui, David Harrington, Dave Thaler, Tassos Chatzithomaoglou, Tom Taylor, Hui Deng, Carlos Pignataro, Matt Miller, Terry Manderson, and other members of the Softwire working group.

Authors’ Addresses

Yu Fu
CNNIC
No.4 South 4th Street, Zhongguancun
Hai-Dian District, Beijing 100190
China
Email: fuyu@cnic.cn

Sheng Jiang
Huawei Technologies Co., Ltd
Q14, Huawei Campus, No.156 Beiqing Road
Hai-Dian District, Beijing 100095
China
Email: jiangsheng@huawei.com

Jiang Dong
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
China
Email: knight.dongjiang@gmail.com

Yuchi Chen
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
China
Email: flashfoxmx@gmail.com