

# **DS-Lite Management Information Base (MIB)**

**draft-fu-softwire-dslite-mib-02**

*Yu Fu*

*Sheng Jiang*

*Yong Cui*

*Jiang Dong*

# Why we need DS-Lite MIB

The NAT-MIB [RFC4008] is designed to carry translation from any address family to any address family, therefore supports IPv4 to IPv4 translation.

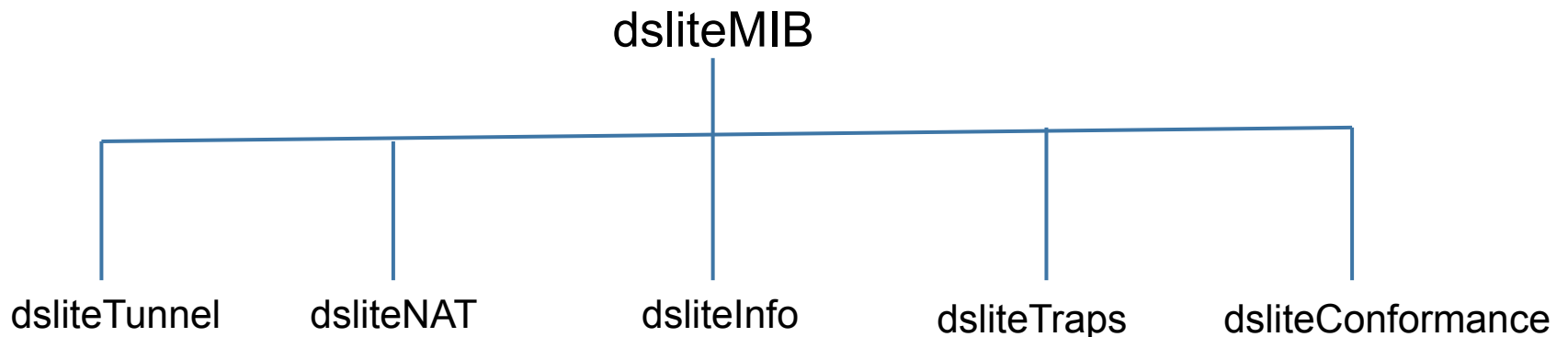
The tunnel MIB [RFC4087] is designed for managing tunnels of any type over IPv4 and IPv6 networks, therefore supports IP in IP tunnels.

**So why we need DS-Lite MIB?**

However, In DS-Lite scenario, the AFTR is not only the tunnel end concentrator, but also a 4-4 translator. **Within the AFTR, tunnel information and translation information MUST be mapped each other.**

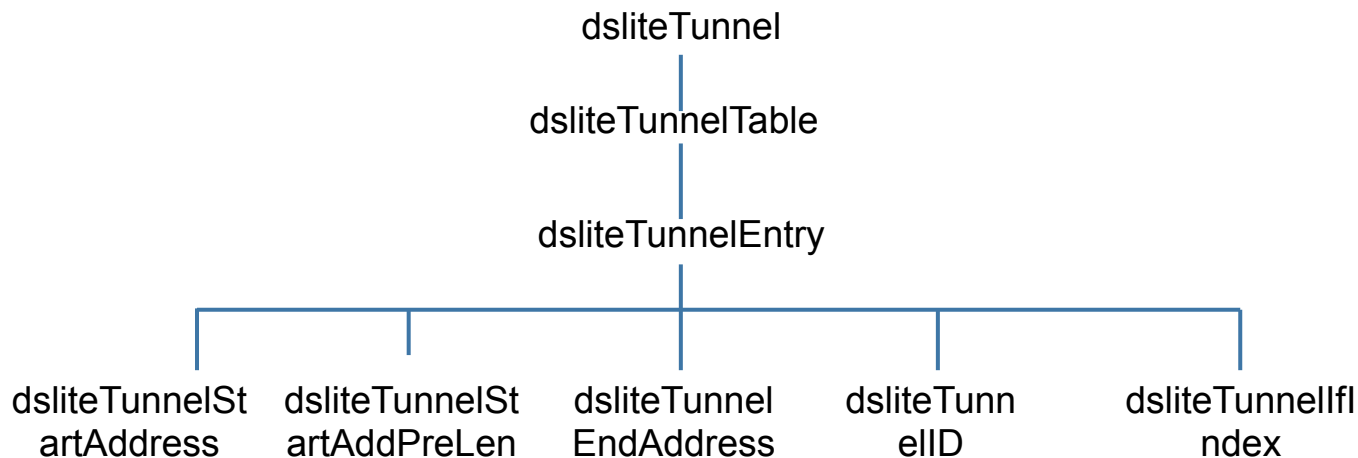
# Subtree of DS-Lite MIB

- **Position of DS-Lite MIB:** `dsliteMIB ::= {transmission xxx}`
- **dsliteTunnel:** Information about Tunnel
- **dsliteNAT:** Information about NAT
- **dsliteInfo:** statistical information in the DS-Lite instance
- **dsliteTraps:** the alarm information
- **dsliteConformance**



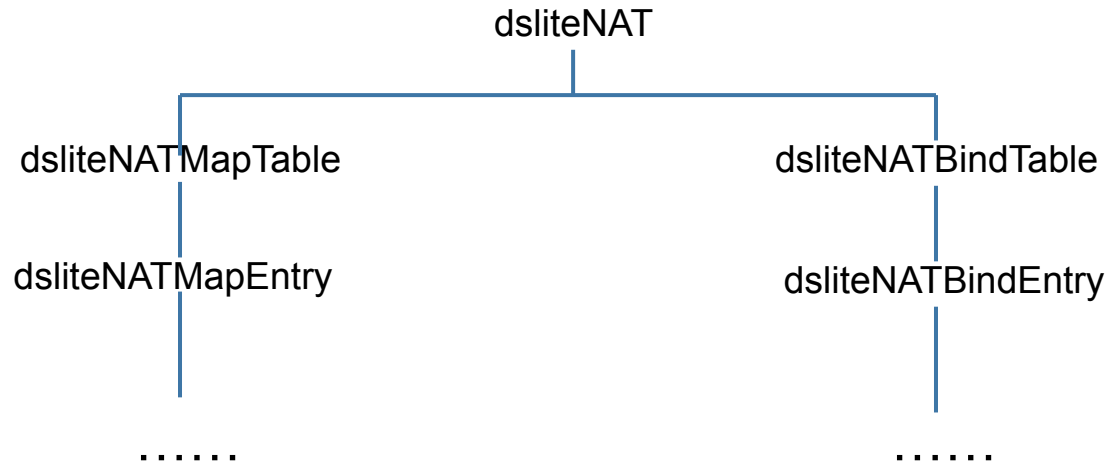
# dsliteTunnel ::= {dsliteMIB 1}

- **DS-Lite Tunnel parameters are defined in this object**
- **Objects imported from Tunnel MIB: dsliteTunnelIndex**
- **New objects defined in our MIB: dsliteTunnelID, dsliteTunnelStartAddress, dsliteTunnelStartAddPreLen, dsliteTunnelEndAddress (Since these objects defined in Tunnel MIB is not access, we define these new objects in our MIB)**



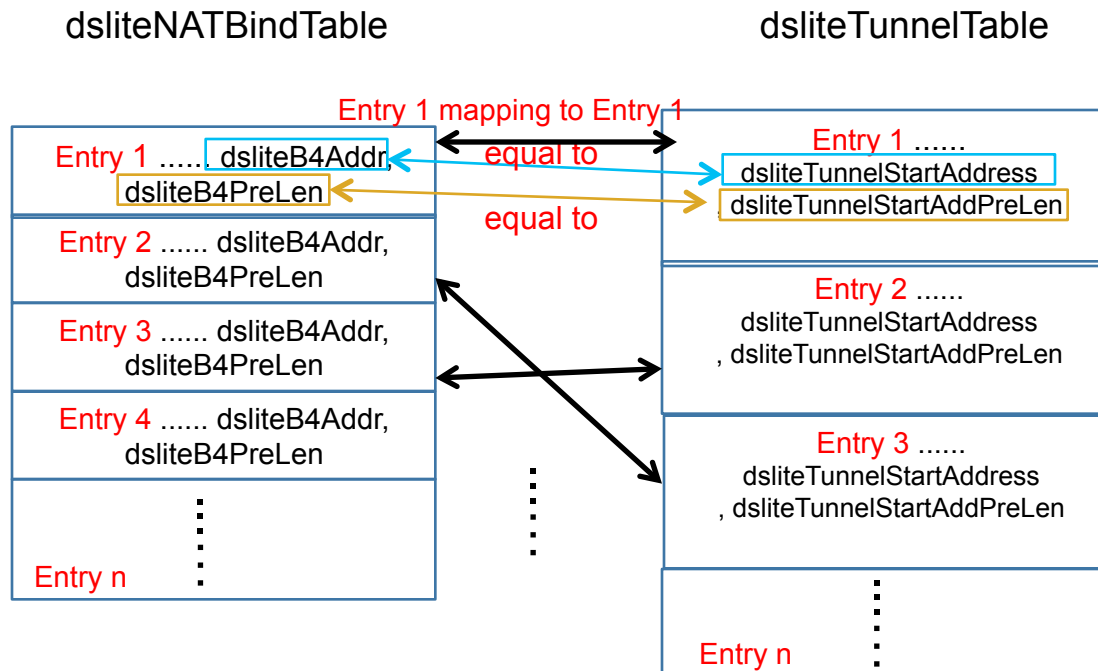
# dsliteNAT ::= { dsliteMIB 2 }

- **DS-Lite NAT parameters are defined in this object**
- **Objects imported from NAT MIB:** 19 objects are imported from NAT MIB
- **New objects defined in our MIB:** dsliteNATBindLocalAddr, dsliteNATBindLocalPort (Since these objects defined in NAT MIB is not access, we define this new objects in our MIB), dsliteB4Addr, dsliteB4PreLen



# Mapping information for Tunnel and NAT

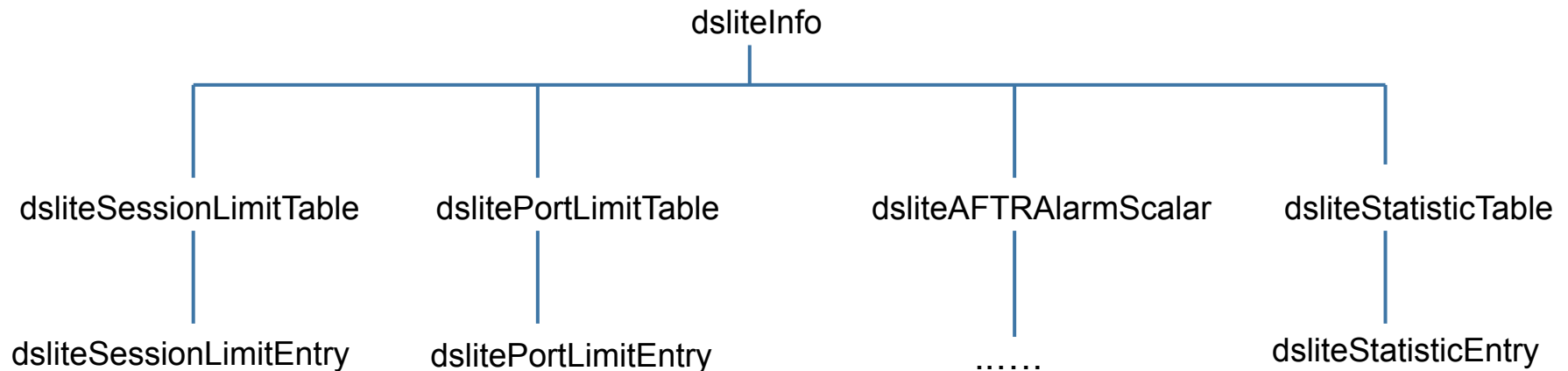
- We define `dsliteTunnelStartAddress`, `dsliteTunnelStartAddPreLen` in `dsliteTunnel` subtree to present the Ipv6 address and prefix for B4. These two objects can be used to identify the tunnel entry. So the `dsliteB4Addr`, `dsliteB4PreLen` in `dsliteNATBindTable` entry are equalled to the these two objects to map the `dsliteTunnelTable` entry



**Open Question:** Another method to create mapping information for Bind Entry is to use the **RowPointer** to point to the Tunnel entry, does that method better?

# dsliteInfo ::= { dsliteMIB 3 }

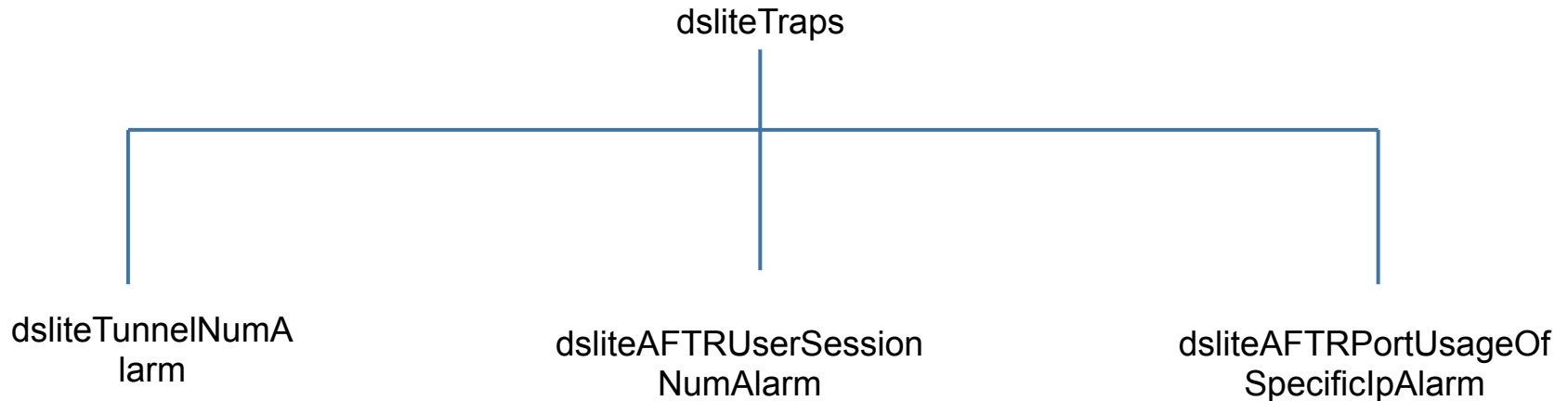
- **The statistical information for DS-lite scenario are described in this object.**
  - The number and type of session limit are defined in this object
  - The number and type of port limit are defined in this object
  - Some alarm scalars of AFTR are defined in this object
  - The statistical parameters for packets and session are defined in this object.



**Open Question:** Shall we need to describe the statistical information in our mib?

# dsliteTraps ::= { dsliteMIB 4 }

- **The Alarm information for the DS-Lite scenario is defined in this object.**



**Open Question:** Shall we need the alarm information in our mib?

Are there any other alarm objects need to define?



**Comments and suggestions are welcome!**

**Adopt as software WG item?**

**Thank you !**