Mobile Ad hoc Networks Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 22, 2011

S. Ratliff
B. Berry
G. Harrison
S. Jury
D. Satterwhite
Cisco Systems
November 22, 2010

Dynamic Link Exchange Protocol (DLEP)
draft-ietf-manet-dlep-00

Abstract

When routing devices rely on modems to effect communications over wireless links, they need timely and accurate knowledge of the characteristics of the link (speed, state, etc.) in order to make forwarding decisions. In mobile or other environments where these characteristics change frequently, manual configurations or the inference of state through routing or transport protocols does not allow the router to make the best decisions. A bidirectional, event-driven communication channel between the router and the modem is necessary.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

There exist today a collection of modem devices that control links of variable bandwidth and quality. Examples of these types of links include line-of-sight (LOS) radios, satellite terminals, and cable/DSL modems. Fluctuations in speed and quality of these links can occur due to configuration (in the case of cable/DSL modems), or on a moment-to-moment basis, due to physical phenomena like multipath interference, obstructions, rain fade, etc. It is also quite possible that link quality and bandwidth varies with respect to individual neighbors on a link, and with the type of traffic being sent. As an example, consider the case of an 802.11g access point, serving 2 associated laptop computers. In this environment, the answer to the question "What is the bandwidth on the 802.11g link?" is "It depends on which associated laptop we're talking about, and on what kind of traffic is being sent." While the first laptop, being physically close to the access point, may have a bandwidth of 54Mbps for unicast traffic, the other laptop, being relatively far away, or obstructed by some object, can simultaneously have a bandwidth of only 32Mbps for unicast. However, for multicast traffic sent from the access point, all traffic is sent at the base transmission rate (which is configurable, but depending on the model of the access point, is usually 24Mbps or less).

In addition to utilizing variable bandwidth links, mobile networks are challenged by the notion that link connectivity will come and go over time. Effectively utilizing a relatively short-lived connection is problematic in IP routed networks, as routing protocols tend to rely on independent timers at OSI Layer 3 to maintain network convergence (e.g. HELLO messages and/or recognition of DEAD routing adjacencies). These short-lived connections can be better utilized with an event-driven paradigm, where acquisition of a new neighbor (or loss of an existing one) is somehow signaled, as opposed to a timer-driven paradigm.

Another complicating factor for mobile networks are the different methods of physically connecting the modem devices to the router. Modems can be deployed as an interface card in a router's chassis, or as a standalone device connected to the router via Ethernet, USB, or even a serial link. In the case of Ethernet or serial attachment, with existing protocols and techniques, routing software cannot be aware of convergence events occurring on the radio link (e.g. acquisition or loss of a potential routing neighbor), nor can the router be aware of the actual capacity of the link. This lack of awareness, along with the variability in bandwidth, leads to a situation where quality of service (QoS) profiles are extremely difficult to establish and properly maintain. This is especially true of demand-based access schemes such as Demand Assigned Multiple Access (DAMA) implementations used on some satellite systems. With a DAMA-based system, additional bandwidth may be available, but will not be used unless the network devices emit traffic at rate higher than the currently established rate. Increasing the traffic rate does not guarantee additional bandwidth will be allocated; rather, it may result in data loss and additional retransmissions on the link.

In attempting to address the challenges listed above, the authors have developed the Data Link Exchange Protocol, or DLEP. The DLEP protocol runs between a router and its attached modem devices, allowing the modem to communicate link characteristics as they change, and convergence events (acquisition and loss of potential routing neighbors). The diagram below is used to illustrate the scope of DLEP sessions. When a local client (Modem device) detects the presence of a remote neighbor, it sends an indication to its local router via the DLEP session. Upon receipt of the indication, the local router would take appropriate action (e.g. initiation of discovery or HELLO protocols) to converge the network. After notification of the new neighbor, the modem device utilizes the DLEP session to report the characteristics of the link (bandwidth, latency, etc) to the router on an as-needed basis. Finally, the Modem is able to use the DLEP session to notify the router when the remote neighbor is lost, shortening the time required to re-converge the network.

```
|-----Local Neighbor-----|          |-----Remote Neighbor----|
|                        |          |     (far-end device)   |

+--------+       +-------+          +-------+       +--------+
| Router |=======| Modem |{~~~~~~~~}| Modem |=======| Router |
|        |       | Device|          | Device|       |        |
+--------+       +-------+          +-------+       +--------+

         |       |       | Link     |       |       |
         |-DLEP--|       | Protocol |       |-DLEP--|
         |       |       | (e.g.    |       |       |
         |       |       | 802.11)  |       |       |
```

Figure 1: DLEP Network

DLEP exists as a collection of type-length-value (TLV) based messages formatted using RFC 5444. The protocol can be used for both Ethernet-attached modems (utilizing, for example, a UDP socket for transport of the RFC 5444 packets), or in environments where the modem is an interface card in a chassis (via a message passing scheme). DLEP utilizes a session paradigm between the modem device and its associated router. If multiple modem devices are attached to a router, a separate DLEP session MUST exist for each modem. If a modem device supports multiple connections to a router (via multiple interfaces), or supports connections to multiple routers, a separate DLEP session MUST exist for each connection.

1.1 Requirements

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 BCP 14, RFC 2119 [RFC2119].

2. Assumptions

In order to implement discovery in the DLEP protocol (thereby avoiding some configuration), we have defined a first-speaker and a passive-listener. Specifically, the router is defined as the passive-listener, and the modem device defined as the first-speaker (e.g. the initiator for discovery). Borrowing from existing terminology, this document refers to the first-speaker as the 'client', even though there is no client/server relationship in the classic sense.

DLEP assumes that participating modem devices appear to the router as a transparent bridge - specifically, the assumption is that the destination MAC address for data traffic in any frame emitted by the router should be the MAC address of the next-hop router or end-device, and not the MAC address of any of the intervening modem devices.

DLEP assumes that security on the session (e.g. authentication of session partners, encryption of traffic, or both) is dealt with by the underlying transport mechanism for the RFC 5444 packets (e.g. by using a transport such as DTLS [DTLS]).

The RFC 5444 message header Sequence Number MUST be included in all DLEP packets. Sequence Numbers start at 1 and are incremented by one for each original and retransmitted message. The unsigned 16-bit Sequence Number rolls over at 65535 to 1. A Sequence Number of 0 is not valid. Peer level Sequence Numbers are unique within the context of a DLEP session. Sequence numbers are used in DLEP to correlate a response to a request.

3. Normal Session Flow

A session between a router and a client is established by exchanging the "Peer Discovery" and "Peer Offer" messages described below.

Once that exchange has successfully occurred, the client informs the router of the presence of a new potential routing partner via the "Neighbor Up" message. The loss of a neighbor is communicated via the "Neighbor Down" message, and link quality is communicated via the "Neighbor Update" message. Note that, due to the issue of metrics varying depending on neighbor (discussed above), DLEP link metrics are expressed within the context of a neighbor relationship, instead of on the link as a whole.

Once the DLEP session has started, the session partners exchange heartbeat messages based on a negotiated time interval. The heartbeat messages are used to assure the session partners are in an appropriate state, and that bidirectional connectivity still exists.

In addition to receiving metrics about the link, DLEP provides for the ability for the router to request a different amount of bandwidth, or latency, for its client via the Link Characteristics Message. This allows the router to deal with requisite increases (or decreases) of allocated bandwidth/latency in demand-based schemes in a more deterministic manner.

4. Generic DLEP Packet Definition

The Generic DLEP Packet Definition follows the format for packets defined in RFC 5444.

The Generic DLEP Packet Definition contains the following fields:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version| Flags | Packet Sequence Number        | Packet TLV    |
|       |       |                               | Block...      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Message...                                                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Version - Version of RFC5444 specification on which the packet/messages/TLVs are constructed.

Flags - 4 bit field. Only bit 1 (phastlv) is set/used. All other bits MUST be ignored by DLEP implementations.

Packet Sequence Number - If present, the packet sequence number is parsed and ignored. DLEP does NOT use or generate packet sequence numbers.

Packet TLV block - a TLV block which contains packet level TLV information.

Message - the packet MAY contain zero or more messages.

5. Generic DLEP Message Format

The Generic DLEP Message Format follows the format for MANET messages defined in RFC 5444. The <msg-seq-num> field, which is OPTIONAL in RFC 5444, MUST exist in all DLEP messages.

The Generic DLEP Message Format contains the following fields:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Msg Type   |Msg Flg|AddrLen|          Message Size         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Message Seq Num      |           TLV Block...        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Message Type – an 8-bit field which specifies the type of the message

Message Flags – Set to 0x1 (bit 3, mhasseqnum bit is set). All other bits are unused and MUST be set to '0'.

Message Address Length – a 4-bit unsigned integer field encoding the length of all addresses included in this message. DLEP implementations do not use this field; contents SHOULD be ignored.

Message Size – a 16-bit unsigned integer field which specifies the number of octets that make up the message including the message header.

Message Sequence Number – a 16-bit unsigned integer field that contains a sequence number, generated by the originator of the message. Sequence numbers range from 1 to 65535. Sequence numbers roll over at 65535 to 1; 0 is invalid.

TLV Block – TLV Block included in the message.

6. Generic DLEP TLV Block Format

The Generic DLEP TLV Block Format follows the format for MANET message TLVs defined in RFC 5444.

The Generic DLEP TLV Block Format contains the following fields:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       TLVs Length             | TLV Type      | TLV Flags     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Length      |                    Value...                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

TLVs Length – a 16-bit unsigned integer field that contains the total number of octets in all of the immediately following TLV elements (tlvs-length not included).

TLV Type - an 8-bit unsigned integer field specifying the type of the TLV.

TLV Flags - an 8-bit flags bit field. Only bit 3 (thasvalue) is set, all other bits are not used and MUST be set to '0'.

Length - Length of the value field of the TLV

Value - A field of length <Length> which contains data specific to a particular TLV type.

7. DLEP TLVs

TLV Value	TLV Description
TBD	Identification TLV
TBD	DLEP Version TLV
TBD	Peer Type TLV
TBD	MAC Address TLV
TBD	IPv4 Address TLV
TBD	IPv6 Address TLV
TBD	Maximum Data Rate (MDR) TLV
TBD	Current Data Rate (CDR) TLV
TBD	Latency TLV
TBD	Resources TLV
TBD	Relative Link Quality (RLQ) TLV
TBD	Status TLV

7.1 Identification TLV

This TLV MUST be in the Packet Header TLV Block for all DLEP messages. It contains client and router identification information used for all messages contained within the packet.

The Identification TLV contains the following fields:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Type = TBD |TLV Flags=0x10 |Length = 9     |  Heartbeat    |
|               |               |               |   Interval    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Router ID                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Client ID                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

TLV Type - Value TBD

TLV Flags - 0x10, Bit 3 (thasvalue) is set, all other bits are unused and MUST be set to '0'.

Length - 9

Heartbeat Interval - An 8-bit, unsigned value containing the heartbeat interval, in seconds for this session. A value of '0' indicates that no heartbeats are used on this session.
This value is used during the Peer Discovery/Peer Offer exchange. In other packets, the value MUST be ignored.
The Heartbeat timer runs at a peer-to-peer level, that is, it runs between a router and a modem device. If a peer does NOT receive any messages for some number of Heartbeat intervals (default 4), it should initiate DLEP session termination procedures.

Router ID - indicates the router ID of the DLEP session.

Client ID - indicates the client ID of the DLEP session.

When the client initiates discovery (via the Peer Discovery message), it MUST set the Client ID to a 32-bit quantity that will be used to uniquely identify this session from the client-side. The client MUST

set the Router ID to '0'. When responding to the Peer Discovery message, the router MUST echo the Client ID, and MUST supply its own unique 32-bit quantity to identify the session from the router's perspective. After the Peer Discovery/Peer Offer exchange, both the Client ID and the Router ID MUST be set to the values obtained from the Peer DIscovery/Peer Offer sequence.

7.2 DLEP Version TLV

The DLEP Version TLV is OPTIONAL, and is used to indicate the client or router version of the protocol. The client and router MAY use this information to decide if the peer is running at a supported level.

The DLEP Version TLV contains the following fields:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Type =TBD  |TLV Flags=0x10 |Length=4       | Major Version |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Major Version |       Minor Version           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

TLV Type - TBD

TLV Flags - 0x10, Bit 3 (thasvalue) is set, all other bits are not used and MUST be set to '0'.

Length - Length is 4

Major Version - Major version of the client or router protocol.

Minor Version - Minor version of the client or router protocol.

Support of this draft is indicated by setting the Major Version to '1', and the Minor Version to '0' (e.g. Version 1.0).

7.3 Peer Type TLV

The Peer Type TLV is used by the router and client to give additional information as to its type. It is an OPTIONAL TLV in both the Peer Discovery Message and the Peer Offer message. The peer type is a string and is envisioned to be used for informational purposes (e.g. display command).

The Peer Type TLV contains the following fields:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Type =TBD   |TLV Flags=0x10 |Length= peer   |Peer Type Str  |
|               |               |type string len|Max Len = 80   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

TLV Type - TBD

TLV Flags - 0x10, Bit 3 (thasvalue) is set, all other bits are not used and MUST be set to '0'.

Length - length of peer type string (80 bytes maximum)
Peer Type String - Non-Null terminated peer type string, maximum length of 80 bytes. For example, a satellite modem might set this variable to 'Satellite terminal'.

7.4 MAC Address TLV

The MAC address TLV MUST appear in all neighbor-oriented messages (e.g. Neighbor Up, Neighbor Up ACK, Neighbor Down, Neighbor Down ACK, Neighbor Update, Link Characteristics Request, and Link Characteristics ACK). The MAC Address TLV contains the address of the far-end (neighbor) router.

The MAC Address TLV contains the following fields:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Type =TBD  |TLV Flags=0x10 |Length = 6     |MAC Address    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      MAC Address                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| MAC Address   |
+-+-+-+-+-+-+-+-+
```

TLV Type   - TBD

TLV Flags   - 0x10, Bit 3 (thasvalue) is set, all other bits are not used and MUST be set to ’0’.

Length   - 6

MAC Address - MAC Address of the far-end router.

7.5 IPv4 Address TLV

The IPv4 Address TLV MAY be used in Neighbor Up, Neighbor Update, and Peer Update Messages, if the client is aware of the Layer 3 address. When included in Neighbor messages, the IPv4 Address TLV contains the IPv4 address of the far-end router (neighbor). In the Peer Update message, it contains the IPv4 address of the local router. In either case, the TLV also contains an indication of whether this is a new or existing address, or is a deletion of a previously known address.

The IPv4 Address TLV contains the following fields:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Type =TBD  |TLV Flags=0x10 |Length = 4     |   Add/Drop    |
|               |               |               |   Indicator   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    IPv4 Address                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

TLV Type   - TBD

TLV Flags   - 0x10, Bit 3 (thasvalue) is set, all other bits are not used and MUST be set to ’0’.

Length   - 5

```
   Add/Drop     - Value indicating whether this is a new or
   Indicator      existing address (0x01), or a withdrawal of
                  an address (0x02).

   IPv4 Address - IPv4 Address of the far-end router.
```

7.6 IPv6 Address TLV

The IPv6 Address TLV MAY be used in Neighbor Up, Neighbor Update, and Peer Update Messages, if the client is aware of the Layer 3 address. When included in Neighbor messages, the IPv6 Address TLV contains the IPv6 address of the far-end router (neighbor). In the Peer Update, it contains the IPv6 address of the local router. In either case, the TLV also contains an indication of whether this is a new or existing address, or is a deletion of a previously known address.

The IPv6 Address TLV contains the following fields:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |TLV Type =TBD  |TLV Flags=0x10 |Length = 17    |   Add/Drop    |
   |               |               |               |   Indicator   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                         IPv6 Address                          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                         IPv6 Address                          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                         IPv6 Address                          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                         IPv6 Address                          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
   TLV Type     - TBD

   TLV Flags    - 0x10, Bit 3 (thasvalue) is set, all other bits are not
                  used and MUST be set to '0'.

   Length       - 17

   Add/Drop     - Value indicating whether this is a new or
   Indicator      existing address (0x01), or a withdrawal of
                  an address (0x02).

   IPv6 Address - IPv6 Address of the far-end router.
```

7.7 Maximum Data Rate TLV

The Maximum Data Rate (MDR) TLV is used in Neighbor Up, Neighbor Update, and Link Characteristics ACK Messages to indicate the maximum theoretical data rate, in bits per second, that can be achieved on the link. When metrics are reported via the messages listed above, the maximum data rate MUST be reported.

The Maximum Data Rate TLV contains the following fields:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Type =TBD  |TLV Flags=0x10 |Length = 8     |  MDR (bps)    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         MDR (bps)                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         MDR (bps)                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
TLV Type                -  TBD

TLV Flags               -  0x10, Bit 3 (thasvalue) is set, all other
                           bits are not used and MUST be set to '0'.

Length                  -  8

Maximum Data Rate       -  A 64-bit unsigned number, representing the
                           maximum theoretical data rate, in bits per
                           second (bps), that can be achieved on the
                           link.
```

7.8 Current Data Rate TLV

The Current Data Rate (CDR) TLV is used in Neighbor Up, Neighbor Update, Link Characteristics Request, and Link Characteristics ACK messages to indicate the rate at which the link is currently operating, or in the case of the Link Characteristics Request, the desired data rate for the link.

The Current Data Rate TLV contains the following fields:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Type =TBD  |TLV Flags=0x10 |Length = 8     |CDR (bps)      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         CDR (bps)                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         CDR (bps)                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
   TLV Type               -  TBD

   TLV Flags              -  0x10, Bit 3 (thasvalue) is set, all other
                             bits are not used and MUST be set to '0'.

   Length                 -  8

   Current Data Rate      -  A 64-bit unsigned number, representing the
                             current data rate, in bits per second (bps),
                             on the link. When reporting metrics (e.g,
                             in Neighbor Up, Neighbor Down, or Link
                             Characteristics ACK), if there is no
                             distinction between current and maximum
                             data rates, current data rate SHOULD be
                             set equal to the maximum data rate.
```

7.9 Latency TLV

The Latency TLV is used in Neighbor Up, Neighbor Update, Link Characteristics Request, and Link Characteristics ACK messages to indicate the amount of latency on the link, or in the case of the Link Characteristics Request, to indicate the maximum latency required (e.g. a should-not-exeed value) on the link.

The Latency TLV contains the following fields:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |TLV Type =TBD  |TLV Flags=0x10 |Length = 2     |Latency (ms)   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |Latency (ms)   |
   +-+-+-+-+-+-+-+-+

   TLV Type               -  TBD

   TLV Flags              -  0x10, Bit 3 (thasvalue) is set, all other
                             bits are not used and MUST be set to '0'.

   Length                 -  2

   Latency                -  the transmission delay that a packet
                             encounters as it is transmitted over the
                             link. In Neighbor Up, Neighbor Update,
                             and Link Characteristics ACK, this value
                             is reported in absolute delay, in
                             milliseconds. The calculation of latency
                             is modem-device dependent. For example,
                             the latency may be a running average
                             calculated from the internal queuing. If
                             the modem device cannot calculate latency,
                             it SHOULD be reported as 0.
```

                              In the Link Characteristics Request Message,
                              this value represents the maximum delay,
                              in milliseconds, expected on the link.

7.10 Resources TLV

The Resources TLV is used in Neighbor Up, Neighbor Update, and Link Characteristics ACK messages to indicate a percentage (0-100) amount of resources (e.g. battery power) remaining on the modem device.

The Resources TLV contains the following fields:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Type =TBD  |TLV Flags=0x10 |Length = 1     |   Resources   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
TLV Type              -  TBD

TLV Flags             -  0x10, Bit 3 (thasvalue) is set, all other
                         bits are not used and MUST be set to '0'.

Length                -  1

Resources             -  a percentage, 0-100, representing the amount
                         of remaining resources, such as battery power.
                         If resources cannot be calculated, a value of
                         100 SHOULD be reported.
```

7.11 Relative Link Quality TLV

The Relative Link Quality (RLQ) TLV is used in Neighbor Up, Neighbor Update, and Link Characteristics ACK messages to indicate the quality of the link as calculated by the modem device.

The Relative Link Quality TLV contains the following fields:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Type =TBD  |TLV Flags=0x10 |Length = 1     |Relative Link  |
|               |               |               |Quality (RLQ)  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
TLV Type              -  TBD

TLV Flags             -  0x10, Bit 3 (thasvalue) is set, all other
                         bits are not used and MUST be set to '0'.

Length                -  1
```

```
   Relative Link Quality -  a non-dimensional number, 0-100,
                            representing the relative link quality.
                            A value of 100 represents a link of the
                            highest quality. If the RLQ cannot be
                            calculated, a value of 100 should be
                            reported.
```

7.12 Status TLV

The Status TLV is sent from either the client or router to indicate the success or failure of a given request

The Status TLV contains the following fields:

```
   0                   1                   2                   3
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |TLV Type =TBD  |TLV Flags=0x10 |Length = 1     |     Code      |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

   TLV Type         - TBD

   TLV Flags        - 0x10, Bit 3 (thasvalue) is set, all other bits are
                      not used and MUST be set to '0'.

   Length           - 1

   Termination Code - 0 = Success
                      Non-zero = Failure. Specific values of a non-
                      zero termination code depend on the operation
                      requested (e.g. Neighbor Up, Neighbor Down, etc).
```

8. DLEP Messages

The DLEP Packet, being based on [RFC5444], contains the following fields:

```
   0                   1                   2                   3
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |Version| Flags |  TLVs Length                  | Identification|
  |  0x0  |  0x2  |                               | TLV Type = TBD|
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |TLV Flags=0x10 | Length = 9    | HB Interval   | Router ID     |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                  Router ID                    | Client ID     |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                  Client ID                    | Message (DLEP |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  | packet can contain zero or more messages)                     |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Version - Version of RFC5444 specification on which the packet/ messages/TLVs are constructed.

Flags - 0x2 Only bit 1 (phastlv) is set/used, all other bits are not used and MUST be set to '0'.

Packet Header TLV Block which contains:
Identification TLV

Message - the packet may contain zero or more messages.

8.1 Message TLVs

TLV Value	TLV Description
TBD	Attached Peer Discovery
TBD	Detached Peer Discovery
TBD	Peer Offer
TBD	Peer Update
TBD	Peer Update ACK
TBD	Peer Termination
TBD	Peer Termination ACK
TBD	Neighbor Up
TBD	Neighbor Up ACK
TBD	Neighbor Down
TBD	Neighbor Down ACK
TBD	Neighbor Update
TBD	Neighbor Address Update
TBD	Neighbor Address Update ACK
TBD	Heartbeat
TBD	Link Characteristics Request
TBD	Link Characteristics ACK

9. Peer Discovery Messages

There are two different types of Peer Discovery Messages, Attached and Detached. Attached Peer Discovery Messages are sent by the client when it is directly attached to the router (e.g. the client exists as a card in the chassis, or it is connected via Ethernet with no intervening devices). The Detached Peer Discovery message, on the other hand, is sent by a "remote" client -- for example, a client at a satellite hub system might use a Detached Discovery Message in order to act as a proxy for remote ground terminals. To explain in another way, a detached client uses the variable link itself (the radio or satellite link) to establish a DLEP session with a remote router.

9.1 Attached Peer Discovery Message

The Attached Peer Discovery Message is sent by an attached client to a router to begin a new DLEP association. The Peer Offer message is required to complete the discovery process. The client MAY implement its own retry heuristics in the event it (the client) determines the Attached Peer Discovery Message has timed out.

The Attached Peer Discovery Message contains the following fields:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Msg Type   |Msg Flg|AddrLen|          Message Size         |
|    = TBD      | 0x1   | 0x3   |15 + size of opt Peer Type TLV |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Message Seq Num      |  TLVs Length = 7 + opt TLVs   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| DLEP Version  |TLV Flags=0x10 | Length = 4    | Major Version |
| TLV Type = TBD|               |               |               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Major Version |         Minor Version         | Peer Type TLV |
|               |                               | Type = TBD    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Flags=0x10 | Length = Len  | Peer Type Str |
|               |of peer string |MaxLen=80 bytes|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Attached Peer Discovery Message - TBD

Message Flags	- Set to 0x1 (bit 3, mhasseqnum bit is set). No other bits are used and MUST be set to '0'.
Message Address Length	- 0x3
Message Size	- 15 + size of optional Peer Type TLV
Message Sequence Number	- a 16-bit unsigned integer field containing a sequence number generated by the message originator.
TLV Block	- TLVs Length: 7 + size of OPTIONAL Peer Type TLV. DLEP Version TLV Peer Type TLV (OPTIONAL)

9.2 Detached Peer Discovery Message

The Detached Peer Discovery Message is sent by a detached client proxy to a router to begin a new DLEP session. The Peer Offer message is required to complete the discovery process. The client MAY implement its own retry heuristics in the event it (the client) determines the Detached Peer Discovery Message has timed out.

The Detached Peer Discovery Message contains the following fields:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Msg Type   |Msg Flg|AddrLen|          Message Size         |
|    = TBD      | 0x1   | 0x3   |15 + size of opt Peer Type TLV |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Message Seq Num      | TLVs Length = 7 + opt TLVs    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| DLEP Version  |TLV Flags=0x10 | Length = 4    | Major Version |
| TLV Type = TBD|               |               |               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Major Version |         Minor Version         | Peer Type TLV |
|               |                               | Type = TBD    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Flags=0x10 | Length = Len  | Peer Type Str |
|               |of peer string |MaxLen=80 bytes|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Detached Peer Discovery Message Type - TBD

Message Flags	- Set to 0x1 (bit 3, mhasseqnum bit is set). All other bits are not used and MUST be set to '0'.
Message Address Length	- 0x3
Message Size	- 15 + size of optional Peer Type TLV
Message Sequence Number	- A 16-bit unsigned integer field containing a sequence number, generated by the message originator.
TLV Block	- TLVs Length: 7 + size of OPTIONAL Peer Type TLV. DLEP Version TLV Peer Type TLV (optional)

10. Peer Offer Message

The Peer Offer Message is sent by a router to a client or client proxy in response to a Peer Discovery Message. The Peer Offer Message is the response to either of the Peer Discovery messages (either Attached or Detached), and completes the DLEP session establishment.

The Peer Offer Message contains the following fields:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Msg Type   |Msg Flg|AddrLen|          Message Size         |
|    = TBD      | 0x1   | 0x3   |15 + size of opt Peer Type TLV |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Message Seq Num      | TLVs Length = 7 + opt TLVs    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| DLEP Version  |TLV Flags=0x10 | Length = 4    | Major Version |
| TLV Type = TBD|               |               |               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Major Version |         Minor Version         | Peer Type TLV |
|               |                               | Type = TBD    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Flags=0x10 | Length = Len  | Peer Type Str |TLV Type = TBD |
|               |of peer string |MaxLen=80 bytes|               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Flags=0x10 |Length = 5     | Add/Drop Ind. | IPv4 Address  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 IPv4 Address                  |TLV Type = TBD |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Flags=0x10 |Length = 17    | Add/Drop Ind. | IPv6 Address  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         IPv6 Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         IPv6 Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         IPv6 Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 IPv6 Address                  |TLV Type = TBD |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Flags=0x10 | Length = 1    | Code          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Peer Offer Message Type - TBD

Message Flags - Set to 0x1 (bit 3, mhasseqnum bit is set). All other bits are unused and MUST be set to '0'.

Message Address Length - 0x3

Message Size - 15 + size of optional Peer Type TLV

Message Sequence Number - A 16-bit unsigned integer field containing a sequence number, generated by the message originator.

TLV Block - TLV Length: 7 + size of optional Peer Type TLV.
DLEP Version TLV
Peer Type TLV (OPTIONAL)
IPv4 Address TLV (OPTIONAL)
IPv6 Address TLV (OPTIONAL)
Status TLV (OPTIONAL)

11. Peer Update Message

The Peer Update message is sent by the router to indicate local Layer 3 address changes. For example, addition of an IPv4 address to the router would prompt a Peer Update message to its attached DLEP clients. If the modem device is capable of understanding and forwarding this information, the address update would prompt any remote DLEP clients (DLEP clients that are on the far-end of the variable link) to issue a "Neighbor Update" message to their local routers, with the address change information. Clients that do not track Layer 3 addresses MUST silently ignore the Peer Update Message. Clients that track Layer 3 addresses MUST acknowledge the Peer Update with a Peer Update ACK message. Routers MAY employ heuristics to retransmit Peer Update messages. Sending of Peer Update Messages SHOULD cease when a router implementation determines that a partner modem device does NOT support Layer 3 address tracking.

The Peer Update Message contains the following fields:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Msg Type   |Msg Flg|AddrLen|          Message Size         |
|    = TBD      | 0x1   | 0x3   |          8 + opt TLVs         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Message Seq Num      | TLVs Length = length of opt   |
|                               |               TLVs            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Type =TBD  |TLV Flags=0x10 |Length = 5     | Add/Drop Ind. |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         IPv4 Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Type =TBD  |TLV Flags=0x10 |Length = 17    | Add/Drop Ind. |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        IPv6 Address                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        IPv6 Address                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        IPv6 Address                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        IPv6 Address                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Peer Update Message Type - TBD

Message Flags - Set to 0x1 (bit 3, mhasseqnum bit is set). All other bits are unused and MUST be set to '0'.

Message Address Length - 0x3

Message Size - 8 + optional TLVs

Message Sequence Number - A 16-bit unsigned integer field containing a sequence number generated by the message originator.

TLV Block - TLV Length: length of optional TLVs.
IPv4 Address TLV (OPTIONAL)
IPv6 Address TLV (OPTIONAL)

12. Peer Update ACK Message

The client sends the Peer Update ACK Message to indicate whether a Peer Update Message was successfully processed.

The Peer Update ACK message contains the following fields:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Msg Type   |Msg Flg|AddrLen|          Message Size         |
|    = TBD      | 0x1   | 0x3   |               12              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Message Seq Num      |  TLVs Length = 4              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Type =TBD  |TLV Flags=0x10 |Length = 1     |      Code     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
Peer Update ACK
Message Type              - TBD

Message Flags             - Set to 0x1 (bit 3, mhasseqnum bit
                            is set). All other bits are unused and
                            MUST be set to '0'.

Message Address Length    - 0x3

Message Size              - 12


Message Sequence Number   - A 16-bit unsigned integer field containing
                            the sequence number from the Neighbor Up
                            Message that is being acknowledged.

TLV Block                 - TLV Length:  4
                            Status TLV
```

13. Peer Termination Message

The Peer Termination Message is sent by either the client or the router when a session needs to be terminated. Transmission of a Peer Termination ACK message is required to confirm the termination process. The sender of the Peer Termination message is free to define its heuristics in event of a timeout. The receiver of a Peer Termination Message MUST terminate all neighbor relationships and release associated resources. No Neighbor Down messages are sent.

The Peer Termination Message contains the following fields:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Msg Type   |Msg Flg|AddrLen|          Message Size         |
|    = TBD      | 0x1   | 0x3   |              12               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Message Seq Num      |           TLVs Length = 4     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Type = TDB |TLV Flags=0x10 | Length = 1    | Code          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Peer Termination Message Type - TBD

Message Flags - Set to 0x1 (bit 3, mhasseqnum bit is set). All other bits are unused and MUST be set to '0'.

Message Address Length - 0x3

Message Size - 12

Message Sequence Number - A 16-bit unsigned integer field containing a sequence number generated by the message originator.

TLV Block - TLV Length = 4. Status TLV

14. Peer Termination ACK Message

The Peer Termination Message ACK is sent by either the client or the router when a session needs to be terminated.

The Peer Termination ACK Message contains the following fields:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Msg Type   |Msg Flg|AddrLen|          Message Size         |
|    = TBD      | 0x1   | 0x3   |              12               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Message Seq Num      |           TLVs Length = 4     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Type = TDB |TLV Flags=0x10 | Length = 1    | Code          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Peer Termination ACK
Message Type - TBD

Message Flags	- Set to 0x1 (bit 3, mhasseqnum bit is set). All other bits are unused and MUST be set to '0'.
Message Address Length	- 0x3
Message Size	- 12
Message Sequence Number	- A 16-bit unsigned integer field containing the sequence number in the corresponding Peer Termination Message being acknowledged.
TLV Block	- TLV Length = 4. Status TLV

15. Neighbor Up Message

The client sends the Neighbor Up message to report that a new potential routing neighbor has been detected. A Neighbor Up ACK Message is required to confirm a received Neighbor Up. The sender of the Neighbor Up Message is free to define its retry heuristics in event of a timeout.

The Neighbor Up Message contains the following fields:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Msg Type   |Msg Flg|AddrLen|          Message Size         |
|    = TBD      | 0x1   | 0x3   |          17 + opt TLVs        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Message Seq Num      |  TLVs Length = 9  + opt TLVs  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Type =TBD  |TLV Flags=0x10 |Length = 6     |MAC Address    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       MAC Address                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| MAC Address   |TLV Type =TBD  |TLV Flags=0x10 |Length = 5     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Add/Drop Ind. |                 IPv4 Address                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|IPv4 Address   |TLV Type =TBD  |TLV Flags=0x10 |Length = 17    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Add/Drop Ind. |        IPv6 Address                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        IPv6 Address                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        IPv6 Address                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        IPv6 Address                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|IPv6 Address   |TLV Type = TBD |TLV Flags=0x10 |Length = 8     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        MDR (bps)                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        MDR (bps)                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Type = TBD |TLV Flags=0x10 |Length = 8     | CDR (bps)     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        CDR (bps)                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        CDR (bps)              |TLV Type = TBD |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Flags=0x10 |Length = 2     |        Latency (ms)           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Type = TBD |TLV Flags=0x10 | Length = 1    |   Resources   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Type = TBD |TLV Flags=0x10 | Length = 1    |      RLQ      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
   Neighbor Up Message Type - TBD

   Message Flags            - Set to 0x1 (bit 3, mhasseqnum bit
                              is set). All other bits are unused and
                              MUST be set to '0'.

   Message Address Length   - 0x3

   Message Size             - 17 + optional TLVs

   Message Sequence Number  - A 16-bit unsigned integer field containing
                              a sequence number generated by the message
                              originator.

   TLV Block                - TLV Length:  9 + optional TLVs.
                              MAC Address TLV (MANDATORY)
                              IPv4 Address TLV (OPTIONAL)
                              IPv6 Address TLV (OPTIONAL)
                              Maximum Data Rate TLV (OPTIONAL)
                              Current Data Rate TLV (OPTIONAL)
                              Latency TLV (OPTIONAL)
                              Resources TLV (OPTIONAL)
                              Relative Link Factor TLV (OPTIONAL)
```

16. Neighbor Up ACK Message

The router sends the Neighbor Up ACK Message to indicate whether a Neighbor Up Message was successfully processed.

The Neighbor Up ACK message contains the following fields:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |    Msg Type   |Msg Flg|AddrLen|          Message Size         |
   |    = TBD      | 0x1   | 0x3   |             21                |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |          Message Seq Num      |  TLVs Length = 13             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |TLV Type =TBD  |TLV Flags=0x10 |Length = 6     |MAC Address    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                       MAC Address                             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | MAC Address   |TLV Type = TBD |TLV Flags=0x10 | Length = 1    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Code          |
   +-+-+-+-+-+-+-+-+

   Neighbor Up ACK
   Message Type             - TBD
```

```
   Message Flags              - Set to 0x1 (bit 3, mhasseqnum bit
                                is set). All other bits are unused and
                                MUST be set to '0'.

   Message Address Length     - 0x3

   Message Size               - 21

   Message Sequence Number    - A 16-bit unsigned integer field containing
                                the sequence number from the Neighbor Down
                                Message that is being acknowledged.

   TLV Block                  - TLV Length:  13
                                MAC Address TLV (MANDATORY)
                                Status TLV (MANDATORY)
```

17. Neighbor Down Message

The client sends the Neighbor Down message to report when a neighbor is no longer reachable from the client. The Neighbor Down message MUST contain a MAC Address TLV. Any other TLVs present MAY be ignored. A Neighbor Down ACK Message is required to confirm the process. The sender of the Neighbor Down message is free to define its retry heuristics in event of a timeout.

The Neighbor Down Message contains the following fields:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |    Msg Type   |Msg Flg|AddrLen|          Message Size         |
   |    = TBD      | 0x1   | 0x3   |       17 + optional TLVs      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |          Message Seq Num      |  TLVs Length =  9 + optional  |
   |                               |                 TLVs          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |TLV Type =TBD  |TLV Flags=0x10 |Length = 6     |MAC Address    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                      MAC Address                              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | MAC Address   |TLV Type = TBD |TLV Flags=0x10 | Length = 1    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Code          |
   +-+-+-+-+-+-+-+-+
```

```
   Neighbor Down Message Type - TBD

   Message Flags                - Set to 0x1 (bit 3, mhasseqnum bit
                                  is set). All other bits are unused and
                                  MUST be set to '0'.

   Message Address Length       - 0x3
```

```
   Message Size               - 17 + optional TLVs

   Message Sequence Number    - A 16-bit unsigned integer field
                                containing a sequence number generated
                                by the message originator.

   TLV Block                  - TLV Length: 9 + optional TLVs
                                MAC Address TLV (MANDATORY)
                                Status TLV (OPTIONAL)
```

18. Neighbor Down ACK Message

The router sends the Neighbor Down ACK Message to indicate whether a Neighbor Down Message was successfully processed.

The Neighbor Down ACK message contains the following fields:

```
   0                   1                   2                   3
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |    Msg Type   |Msg Flg|AddrLen|          Message Size         |
  |    = TBD      | 0x1   | 0x3   |              21               |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |          Message Seq Num      |  TLVs Length = 13             |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |TLV Type =TBD  |TLV Flags=0x10 |Length = 6     |MAC Address    |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                         MAC Address                           |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  | MAC Address   |TLV Type = TBD |TLV Flags=0x10 | Length = 1    |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  | Code          |
  +-+-+-+-+-+-+-+-+
```

```
   Neighbor Down ACK
   Message Type             - TBD

   Message Flags            - Set to 0x1 (bit 3, mhasseqnum bit
                              is set). All other bits are unused and
                              MUST be set to '0'.

   Message Address Length   - 0x3

   Message Size             - 21

   Message Sequence Number  - A 16-bit unsigned integer field containing
                              the sequence number from the Neighbor Down
                              Message that is being acknowledged.

   TLV Block                - TLV Length:  13
                              MAC Address TLV (MANDATORY)
                              Status TLV (MANDATORY)
```

19. Neighbor Update Message

The client sends the Neighbor Update message when a change in link metric parameters is detected for a routing neighbor.

The Neighbor Update Message contains the following fields:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Msg Type   |Msg Flg|AddrLen|          Message Size         |
|    = TBD      | 0x1   | 0x3   |          17 + opt TLVs        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Message Seq Num      |  TLVs Length = 9 + opt TLVs   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Type =TBD  |TLV Flags=0x10 |Length = 6     |MAC Address    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      MAC Address                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| MAC Address   |TLV Type = TBD |TLV Flags=0x10 |Length = 8     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        MDR (bps)                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        MDR (bps)                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Type = TBD |TLV Flags=0x10 |Length = 8     | CDR (bps)     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        CDR (bps)                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        CDR (bps)              |TLV Type = TBD |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Flags=0x10 |Length = 2     |        Latency (ms)           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Type = TBD |TLV Flags=0x10 | Length = 1    |   Resources   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Type = TBD |TLV Flags=0x10 | Length = 1    |      RLQ      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Neighbor Update Message Type - TBD

```
Message Flags                 - Set to 0x1 (bit 3, mhasseqnum
                                bit is set).  All other bits are
                                unused and MUST be set to '0'.

Message Address Length        - 0x3

Message Size                  - 17 + optional TLVs

Message Sequence Number       - A 16-bit unsigned integer field
                                containing a sequence number,
                                generated by the message originator.
```

```
   TLV Block                      - TLVs Length - 9 + optional TLVs.
                                    MAC Address TLV (MANDATORY)
                                    Maximum Data Rate TLV (OPTIONAL)
                                    Current Data Rate TLV (OPTIONAL)
                                    Latency TLV (OPTIONAL)
                                    Resources TLV (OPTIONAL)
                                    Relative Link Quality TLV (OPTIONAL)
```

20. Neighbor Address Update Message

The client sends the Neighbor Address Update message when a change in Layer 3 addressing is detected for a routing neighbor.

The Neighbor Address Update Message contains the following fields:

```
   0                   1                   2                   3
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |    Msg Type   |Msg Flg|AddrLen|          Message Size         |
  |    = TBD      | 0x1   | 0x3   |          17 + opt TLVs        |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |          Message Seq Num      |  TLVs Length = 9 + opt TLVs   |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |TLV Type =TBD  |TLV Flags=0x10 |Length = 6     |MAC Address    |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                        MAC Address                            |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  | MAC Address   |TLV Type =TBD  |TLV Flags=0x10 |Length = 5     |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  | Add/Drop Ind. |                IPv4 Address                   |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |IPv4 Address   |TLV Type =TBD  |TLV Flags=0x10 |Length = 17    |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  | Add/Drop Ind. |        IPv6 Address                           |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                        IPv6 Address                           |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                        IPv6 Address                           |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                        IPv6 Address                           |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
   Neighbor Address Update
   Message Type                   - TBD

   Message Flags                  - Set to 0x1 (bit 3, mhasseqnum
                                    bit is set).  All other bits are
                                    unused and MUST be set to '0'.

   Message Address Length         - 0x3
```

```
   Message Size                 - 17 + optional TLVs

   Message Sequence Number      - A 16-bit unsigned integer field
                                  containing a sequence number,
                                  generated by the message originator.

   TLV Block                    - TLVs Length - 9 + optional TLVs.
                                  MAC Address TLV (MANDATORY)
                                  IPv4 Address TLV (OPTIONAL)
                                  IPv6 Address TLV (OPTIONAL)
```

21. Neighbor Address Update ACK Message

The router sends the Neighbor Address Update ACK Message to indicate whether a Neighbor Address Update Message was successfully processed.

The Neighbor Address Update ACK message contains the following fields:

```
   0                   1                   2                   3
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |    Msg Type   |Msg Flg|AddrLen|          Message Size         |
  |    = TBD      | 0x1   | 0x3   |               21              |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |          Message Seq Num      |  TLVs Length = 13             |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |TLV Type =TBD  |TLV Flags=0x10 |Length = 6     |MAC Address    |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                        MAC Address                            |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  | MAC Address   |TLV Type = TBD |TLV Flags=0x10 | Length = 1    |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  | Code          |
  +-+-+-+-+-+-+-+-+
```

```
   Neighbor Address Update
   ACK Message Type         - TBD

   Message Flags            - Set to 0x1 (bit 3, mhasseqnum bit
                              is set). All other bits are unused and
                              MUST be set to '0'.

   Message Address Length   - 0x3

   Message Size             - 21

   Message Sequence Number  - A 16-bit unsigned integer field containing
                              the sequence number from the Neighbor Down
                              Message that is being acknowledged.
```

```
   TLV Block                 - TLV Length:  13
                               MAC Address TLV (MANDATORY)
                               Status TLV (MANDATORY)
```

22. Heartbeat Message

A Heartbeat Message is sent by a peer every N seconds, where N is defined in the "Heartbeat Interval" field of the discovery message. The message is used by peers to detect when a DLEP session partner is no longer communicating. Peers SHOULD allow some integral number of heartbeat intervals (default 4) to expire with no traffic on the session before initiating DLEP session termination procedures.

The Heartbeat Message contains the following fields:

```
   0                   1                   2                   3
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |    Msg Type    |Msg Flg|AddrLen|       Message Size = 8        |
  |    = TBD       | 0x1   | 0x3   |                               |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |          Message Seq Num      |  TLVs Length = 0              |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
   Message Type            - TBD

   Message Flags           - Set to 0x1 (bit 3, mhasseqnum bit is
                             set). All other bits are unused and SHOULD
                             be set to '0'.

   Message Address Length  - 0x3

   Message Size            - 8

   Message Sequence Number - A 16-bit unsigned integer field containing
                             a sequence number generated by the message
                             originator.
   TLV Block -               TLV Length = 0
```

23. Link Characteristics Request Message

The Link Characteristics Request Message is sent by the router to the modem device when the router detects that a different set of transmission characteristics is necessary (or desired) for the type of traffic that is flowing on the link. The request contains either a Current Data Rate (CDR) TLV to request a different amount of bandwidth than what is currently allocated, a Latency TLV to request that traffic delay on the link not exceed the specified value, or both. A Link Characteristics ACK Message is required to complete the request. Implementations are free to define their retry heuristics in event of a timeout. Issuing a Link Characteristics Request with ONLY the MAC Address TLV is a mechanism a peer MAY use to request metrics (via the Link

Characteristics ACK) from its partner.

The Link Characteristics Request Message contains the following fields:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Msg Type   |Msg Flg|AddrLen|   Message Size = 8 + Length   |
|    = TBD      | 0x1   | 0x3   |   of optional TLVs            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Message Seq Num      |         TLVs Length =         |
|                               |       9 + Optional TLVs       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Type =TBD  |TLV Flags=0x10 |Length = 6     |MAC Address    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      MAC Address                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| MAC Address   |TLV Type =TBD  |TLV Flags=0x10 |Length = 8     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        CDR (bps)                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        CDR (bps)                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Type = TBD |TLV Flags=0x10 |Length = 2     | Latency (ms)  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Latency (ms)  |
+-+-+-+-+-+-+-+-+
```

Message Type   - TBD

Message Flags   - Set to 0x1 (bit 3, mhasseqnum bit is set). All other bits are unused and MUST be set to '0'.

Message Address Length - 0x3

Message Size   - 8 + length of optional (Current Data Rate and/or Latency) TLVs

Message Sequence Number - A 16-bit unsigned integer field containing a sequence number generated by the message originator.

TLV Block   - TLVs Length

MAC Address TLV (MANDATORY)

Current Data Rate TLV - if present, this value represents the requested data rate in bits per second (bps). (OPTIONAL)

Latency TLV - if present, this value represents the maximum latency, in milliseconds, desired on the link. (OPTIONAL)

24. Link Characteristics ACK Message

The Link Characteristics ACK Message is sent by the client to the router letting the router know the success (or failure) of the requested change in link characteristics. The Link Characteristics ACK message SHOULD contain a complete set of metric TLVs. It MUST contain the same TLV types as the request. The values in the metric TLVs in the Link Characteristics ACK message MUST reflect the link characteristics after the request has been processed.

The Link Characteristics ACK Message contains the following fields:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Msg Type   |Msg Flg|AddrLen|   Message Size = 8 + Length   |
|    = TBD      | 0x1   | 0x3   |   of additional TLVs          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Message Seq Num      |  TLVs Length = 9 + optional   |
|                               |                 TLVs          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Type =TBD  |TLV Flags=0x10 |Length = 6     |MAC Address    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        MAC Address                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| MAC Address   |TLV Type =TBD  |TLV Flags=0x10 |Length = 8    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        MDR (bps)                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        MDR (bps)                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Type = TBD |TLV Flags=0x10 |Length = 8     | CDR (bps)     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        CDR (bps)                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        CDR (bps)              |TLV Type = TBD |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Flags=0x10 |Length = 2     |        Latency (ms)           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Type = TBD |TLV Flags=0x10 | Length = 1    |   Resources   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TLV Type = TBD |TLV Flags=0x10 | Length = 1    |      RLQ      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Message Type - TBD

Message Flags - Set to 0x1 (bit 3, mhasseqnum bit is set). All other bits are unused and MUST be set to '0'.

Message Address Length - 0x3

Message Size - 8 + length of optional (Current Data Rate and/or Latency) TLVs

Message Sequence Number - A 16-bit unsigned integer field containing the sequence number that appeared on the corresponding Link Characteristics Request message.

TLV Block - TLVs Length = 9 + Optional TLVs

MAC Address TLV (MANDATORY)

Maximum Data Rate TLV (OPTIONAL)

Current Data Rate TLV - if present, this value represents the NEW (or unchanged, if the request is denied) Current Data Rate in bits per second (bps). (OPTIONAL)

Latency TLV - if present, this value represents the NEW maximum latency (or unchanged, if the request is denied), expressed in milliseconds, on the link. (OPTIONAL)

Resources TLV (OPTIONAL)

Relative Link Quality TLV (OPTIONAL)

25. Security Considerations

The protocol does not contain any mechanisms for security (e.g. authentication or encryption). The protocol assumes that any security would be implemented in the underlying transport (for example, by use of DTLS or some other mechanism), and is therefore outside the scope of this document.

26. IANA Considerations

This section specifies requests to IANA.

26.1 TLV Registrations

This specification defines:

o Twelve TLV types which must be allocated from the 0-223 range of the "Assigned Packet TLV Types" repository of [RFC5444].

- o Seventeen Message types which must be allocated from the 0-127 range of the "Assigning Message TLV Types" repository of [RFC5444].

26.2 Expert Review: Evaluation Guidelines

For the registries for TLV type extensions where an Expert Review is required, the designated expert SHOULD take the same general recommendations into consideration as are specified by [RFC5444].

26.3 Packet TLV Type Registrations

The Packet TLVs specified below must be allocated from the "Packet TLV Types" namespace of [RFC5444].

- o Identification TLV
- o DLEP Version TLV
- o Peer Type TLV
- o MAC Address TLV
- o IPv4 Address TLV
- o IPv6 Address TLV
- o Maximum Data Rate TLV
- o Current Data Rate TLV
- o Latency TLV
- o Resources TLV
- o Relative Link Quality TLV
- o Status TLV

26.4 Message TLV Type Registrations

The Message TLVs specified below must be allocated from the "Message TLV Types" namespace of [RFC5444].

- o Attached Peer Discovery Message
- o Detached Peer Discovery Message
- o Peer Offer Message
- o Peer Update Message
- o Peer Update ACK Message
- o Peer Termination Message
- o Peer Termination ACK Message
- o Neighbor Up Message
- o Neighbor Up ACK Message
- o Neighbor Down Message
- o Neighbor Down ACK Message
- o Neighbor Update Message
- o Neighbor Address Update Message
- o Neighbor Address Update ACK Message
- o Heartbeat Message
- o Link Characteristics Request Message
- o Link Characteristics ACK Message

27. Appendix A.

Peer Level Message Flows

*Modem Device (Client) Restarts Discovery

```
   Router                    Client   Message Description
   ====================================================================

   <-------Peer Discovery---------    Modem initiates discovery


    ---------Peer Offer----------->   Router detects a problem, sends
      w/ Non-zero Status TLV          Peer Offer w/ Status TLV indicating
                                      the error.

                                      Modem accepts failure, restarts
                                      discovery process.

   <-------Peer Discovery---------    Modem initiates discovery


    ---------Peer Offer----------->   Router accepts, sends Peer Offer
         w/ Zero Status TLV           w/ Status TLV indicating success.

                                      Discovery completed.
```

*Modem Device Detects Peer Offer Timeout

```
   Router                    Client   Message Description
   ====================================================================

   <-------Peer Discovery---------    Modem initiates discovery,
                                      starts a guard timer.

                                      Modem guard timer expires.
                                      Modem restarts discovery process.

    <-------Peer Discovery---------   Modem initiates discovery,
                                      starts a guard timer.

    ---------Peer Offer----------->   Router accepts, sends Peer Offer
         w/ Zero Status TLV           w/ Status TLV indicating success.

                                      Discovery completed.
```

*Router Peer Offer Lost

```
   Router                     Client   Message Description
   ====================================================================

   <-------Peer Discovery---------     Modem initiates discovery,
                                       starts a guard timer.

    ---------Peer Offer-------||       Router offers availability

                                       Modem times out on Peer Offer,
                                       restarts discovery process.

   <-------Peer Discovery---------     Modem initiates discovery

    ---------Peer Offer----------->    Router detects subsequent discovery,
                                       internally terminates the previous,
                                       accepts the new association, sends
                                       Peer Offer w/ Status TLV indicating
                                       success.


                                       Discovery completed.
```

*Discovery Success

```
   Router                     Client   Message Description
   ====================================================================

   <-------Peer Discovery---------     Modem initiates discovery

    ---------Peer Offer----------->    Router offers availability

    -------Peer Heartbeat--------->

   <-------Peer Heartbeat---------

    -------Peer Heartbeat--------->

   <==============================>    Neighbor Sessions

   <-------Peer Heartbeat---------

    -------Peer Heartbeat--------->

    --------Peer Term Req--------->    Terminate Request

   <--------Peer Term Res---------     Terminate Response
```

*Router Detects a Heartbeat timeout

```
   Router                    Client   Message Description
   ====================================================================

   <-------Peer Heartbeat---------

    -------Peer Heartbeat--------->

      ||---Peer Heartbeat---------

           ~ ~ ~ ~ ~ ~ ~

    -------Peer Heartbeat--------->

      ||---Peer Heartbeat---------
                                      Router Heartbeat Timer expires,
                                      detects missing heartbeats. Router
                                      takes down all neighbor sessions
                                      and terminates the Peer association.

    ------Peer Terminate --------->   Peer Terminate Request

                                      Modem takes down all neighbor
                                      sessions, then acknowledges the
                                      Peer Terminate

   <----Peer Terminate ACK---------   Peer Terminate ACK
```

*Modem Detects a Heartbeat timeout

```
   Router                    Client   Message Description
   ====================================================================

   <-------Peer Heartbeat---------

    -------Peer Heartbeat------||

   <-------Peer Heartbeat---------

           ~ ~ ~ ~ ~ ~ ~

    -------Peer Heartbeat------||

   <-------Peer Heartbeat---------
                                      Modem Heartbeat Timer expires,
                                      detects missing heartbeats. Modem
                                      takes down all neighbor sessions
                                      and terminates the Peer association.
```

```
    <-------Peer Terminate--------    Peer Terminate Request

                                      Router takes down all neighbor
                                      sessions, then acknowledges the
                                      Peer Terminate

    ------Peer Terminate ACK----->    Peer Terminate ACK
```

*Peer Terminate (from Modem) Lost

```
   Router                    Client   Message Description
   ====================================================================

      ||------Peer Terminate--------   Modem Peer Terminate Request

                                      Router Heartbeat times out,
                                      terminates association.

    --------Peer Terminate------->    Router Peer Terminate

    <-----Peer Terminate ACK------    Modem sends Peer Terminate ACK
```

*Peer Terminate (from router) Lost

```
   Router                    Client   Message Description
   ====================================================================

    -------Peer Terminate-------->    Router Peer Terminate Request

                                      Modem HB times out,
                                      terminates association.

    <------Peer Terminate--------     Modem Peer Terminate

    ------Peer Terminate ACK----->    Peer Terminate ACK
```

Neighbor Level Message Flows

*Modem Neighbor Up Lost

```
   Router                    Client   Message Description
   ====================================================================

    ||-----Neighbor Up ------------   Modem sends Neighbor Up

                                      Modem timesout on ACK

    <------Neighbor Up ------------   Modem sends Neighbor Up

    ------Neighbor Up ACK--------->   Router accepts the neighbor
                                      session

   <------Neighbor Update---------    Modem Neighbor Metrics
           . . . . . . . .
   <------Neighbor Update---------    Modem Neighbor Metrics
```

*Router Detects Duplicate Neighbor Ups

```
   Router                    Client   Message Description
   ====================================================================

    <------Neighbor Up ------------   Modem sends Neighbor Up

    ------Neighbor Up ACK-------||    Router accepts the neighbor
                                      session

                                      Modem timesout on ACK

    <------Neighbor Up ------------   Modem resends Neighbor Up

                                      Router detects duplicate
                                      Neighbor, takes down the
                                      previous, accepts the new
                                      Neighbor.

    ------Neighbor Up ACK--------->   Router accepts the neighbor
                                      session

   <------Neighbor Update---------    Modem Neighbor Metrics
           . . . . . . . .
   <------Neighbor Update---------    Modem Neighbor Metrics
```

*Neighbor Up, No Layer 3 Addresses

```
   Router                    Client    Message Description
   ====================================================================

    <-------Neighbor Up ------------   Modem sends Neighbor Up

    ------Neighbor Up ACK--------->   Router accepts the neighbor
                                       session

                                       Router ARPs for IPv4 if defined.
                                       Router drives ND for IPv6 if
                                       defined.

   <------Neighbor Update---------    Modem Neighbor Metrics
          . . . . . . . .
   <------Neighbor Update---------    Modem Neighbor Metrics
```

*Neighbor Up with IPv4, No IPv6

```
   Router                    Client   Message Description
   ====================================================================

    <------Neighbor Up ------------   Modem sends Neighbor Up with
                                      the IPv4 TLV

    ------Neighbor Up ACK--------->   Router accepts the neighbor
                                      session

                                      Router drives ND for IPv6 if
                                      defined.

   <------Neighbor Update---------    Modem Neighbor Metrics
          . . . . . . . .
   <------Neighbor Update---------    Modem Neighbor Metrics
```

*Neighbor Up with IPv4 and IPv6

```
   Router                    Client   Message Description
   ====================================================================

    <------Neighbor Up ------------   Modem sends Neighbor Up with
                                      the IPv4 and IPv6 TLVs

    ------Neighbor Up ACK--------->   Router accepts the neighbor
                                      session

   <------Neighbor Update---------    Modem Neighbor Metrics
          . . . . . . . .
   <------Neighbor Update---------    Modem Neighbor Metrics
```

*Neighbor Session Success

```
   Router                     Client   Message Description
   ====================================================================

    ---------Peer Offer----------->   Router offers availability

    -------Peer Heartbeat--------->


   <------Neighbor Up -----------      Modem

    ------Neighbor Up ACK-------->     Router

   <------Neighbor Update---------     Modem
          . . . . . . . . .
   <------Neighbor Update---------     Modem

                                       Modem initiates the terminate

   <------Neighbor Down ----------     Modem

    ------Neighbor Down ACK------->    Router

                                       or

                                       Router initiates the terminate

    ------Neighbor Down ---------->    Router

   <------Neighbor Down ACK-------     Modem
```

Acknowledgements

The authors would like to acknowledge the influence and contributions of Chris Olsen and Teco Boot.

Normative References

[RFC5444] Clausen, T., Ed,. "Generalized Mobile Ad Hoc Network (MANET) Packet/Message Format", RFC 5444, Februar, 2009.

[RFC5578] Berry, B., Ed., "PPPoE with Credit Flow and Metrics", RFC 5578, February 2010.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.

Informative References

[DTLS] Rescorla, E., Ed,. "Datagram Transport Layer Security", RFC 4347, April 2006.

An open source (MIT License) DLEP implementation is available at http://sourceforge.net/projects/dleptools

Author's Addresses

Stan Ratliff
Cisco
170 West Tasman Drive
San Jose, CA 95134
USA
EMail: sratliff@cisco.com

Bo Berry
Cisco
170 West Tasman Drive
San Jose, CA 95134
USA
EMail: boberry@cisco.com

Greg Harrison
Cisco
170 West Tasman Drive
San Jose, CA 95134
USA
EMail: greharri@cisco.com

Shawn Jury
Cisco
170 West Tasman Drive
San Jose, CA 95134
USA
Email: sjury@cisco.com

Darryl Satterwhite
Cisco
170 West Tasman Drive
San Jose, CA 95134
USA
Email: dsatterw@cisco.com

Internet Engineering Task Force   S. Harnedy
Internet-Draft   Booz Allen Hamilton
Intended status: Standards Track   R. Cole
Expires: July 23, 2011   US Army CERDEC
                                                            I. Chakeres
                                                                 CenGen
                                                       January 19, 2011

Definition of Managed Objects for the DYMO Manet Routing Protocol
draft-ietf-manet-dymo-mib-04

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 objects for configuring aspects of the DYMO routing process. The DYMO-MIB also reports state information, performance information, and notifications. In addition to configuration, this additional state, performance and notification information is useful to management operators troubleshooting DYMO routing problems.

Table of Contents

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 configuring aspects of a Dynamic MANET On-demand (DYMO) routing [I-D.ietf-manet-dymo] process. The DYMO-MIB also reports state information, performance metrics, and notifications. In addition to configuration, this additional state, performance and notification information is useful to management stations troubleshooting routing problems.

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. Conventions

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 [RFC2119].

4. Overview

The Dynamic MANET On-demand (DYMO) routing protocol [I-D.ietf-manet-dymo] is intended for use by mobile nodes in wireless, multihop networks. DYMO determines unicast routes among DYMO routers within the network in an on-demand fashion, offering improved convergence in dynamic topologies.

A DYMO router's MIB contains DYMO process configuration parameters (e.g. interfaces), state information (e.g. sequence number), performance counters (e.g. number of control messages), and notifications.

4.1. DYMO Management Model

This section describes the management model for the DYMO routing protocol.

The MIB is comprised of four groups, i.e., Notifications, Configuration, State and Performance. The configuration of the managed devices is controlled by the objects in the Configuration Group. These are divided into Nodal and Interface objects. The bulk of the DYMO configuration is in the Nodal objects which control protocol behavior. The Interface objects merely identify/configure interfaces to enable DYMO routing over their interface. The Nodal objects are further divided into routing (or protocol) objects and Gateway objects. Gateway objects define other routing prefixes for which the node acts as a routing proxy on behalf of these non-local prefixes.

The Configuration Objects drive the behavior of the managed DYMO device and hence determines the information in the remaining groups, i.e., State, Performance and Notifications. The State objects primarily present the resulting forwarding table objects. The Performance group primarily is comprised of counters for monitoring the number of DYMO routing messages received locally, per node and per interface. The Notifications group contains objects which monitor changes to the interface configuration and the gateway prefixes configuration.

See the below diagram outlining the DYMO-MIB device management model.

```
   +-------------------------------------------+
   |  CONFIGURATION GROUP                      |
   |                                           |
   |  Nodal                       Interface    |
   |                                           |
   |  +--------+  +--------+      +--------+   |
   |  |Gateway|  |Routing|       |Routing|    |
   |  +--------+  +--------+      +--------+   |
   |                                           |
   +-------------------------------------------+
        ||           ||               ||
        ||           ||               ||
        ||           ||               ||
      \  /         \  /             \  /
       \/           \/               \/
   +--------+  +--------------+  +----------------+
   | STATE  |  | PERFORMANCE  |  | NOTIFICATIONS  |
   | GROUP  |  | GROUP        |  | GROUP          |
   +--------+  +--------------+  +----------------+
```

4.2. Terms

The following definitions apply throughout this document:

- o Configuration Objects - switches, tables, objects which are initialized to default settings or set through the management interface defined by this MIB.

- o Tunable Configuration Objects - objects whose values affect timing or attempt bounds on the DYMO protocol.

- o State Objects - automatically generated values which define the current operating state of the DYMO protocol process in the router.

- o Performance Objects - automatically generated values which help an operator or automated tool to assess the performance of the DYMO protocol process on the router and the overall routing performance within the DYMO routing domain.

5. Structure of the MIB Module

This section presents the structure of the DYMO MIB module. The objects are arranged into the following groups:

- o dymoMIBNotifications - defines the notifications associated with the DYMO-MIB. These are currently limited to notifications of interface state changes and gateway prefix changes.

- o dymoMIBObjects - defines the objects forming the basis for the DYMO-MIB. These objects are divided up by function into the following groups:

- o

  * Configuration Group - This group contains the DYMO objects that configure specific options that determine the overall performance and operation of the routing protocol for the router device and its interfaces.

  * State Group - Contains information describing the current state of the DYMO process such as the DYMO routing table.

  * Performance Group - Contains objects which help to characterize the performance of the DYMO process, typically statistics counters. There are two types of DYMO statistics: global counters and per interface counters.

- o dymoMIBConformance - defines minimal and full conformance of implementations to this DYMO-MIB.

5.1. Textual Conventions

The textual conventions used in the DYMO-MIB are as follows. The RowStatus and TruthValue textual conventions are imported from RFC 2579 [RFC2579]. The DymoInterfaceOperStatus is defined within the DYMO-MIB. This contains the current operational status of the DYMO interface.

5.2. The Configuration Group

The DYMO device is configured with a set of controls. The list of configuration controls for the DYMO device follow.

Protocol Configuration Parameters:

- o DID

- o MSG_HOPLIMIT

- o ROUTE_TIMEOUT

- o ROUTE_AGE_MIN_TIMEOUT

- o ROUTE_SEQNUM_AGE_MAX_TIMEOUT

- ROUTE_USED_TIMEOUT

- ROUTE_DELETE_TIMEOUT

- ROUTE_RREQ_WAIT_TIME

- UNICAST_MESSAGE_SENT_TIMEOUT

- MSG_HOPLIMIT

- DISCOVERY_ATTEMPTS_MAX

Protocol Configuration Tables:

- Responsible Hosts - If RESPONSIBLE_ADDRESSES is set to other than self address, then the DYMO router must be configured with the set of host addresses for which it is to generate RREP messages.

- Interfaces - If DYMO_INTERFACES is set to other than all, then the DYMO router must be told which interfaces to run the DYMO protocol over. This is a table containing the interfaces and associated information.

5.3. The State Group

The State Subtree reports current state information. State information from the DYMO-MIB is primarily contained in the 'Routing' Table.

5.3.1. Routing Table

The DYMO routing table contains information related to IP forwarding entries found by the node's DYMO processes.

5.4. The Performance Group

The Performance subtree reports primarily counters that relate to DYMO protocol activity. The DYMO performance objects consists of per node and per interface objects:

- OwnSequenceNumber

- RREQ initiated

- RREQ sent

- RREQ received

- RREP initiated

- RREP sent

- RREP received

- RRER initiated

- RRER sent

- RRER received

- Per interface statistics table with the following entries:

-

  * RREQ initiated

  * RREQ sent

  * RREQ received

  * RREP initiated

  * RREP sent

  * RREP received

  * RRER initiated

  * RRER sent

  * RRER received

5.5. The Notifications Group

The Notifications Subtree contains the list of notifications supported within the DYMO-MIB and their intended purpose or utility. This group is currently contains two notification objects, one related to status changes in DYMO interfaces and one related to changes in the gateway prefixes table.

6. Relationship to Other MIB Modules

The text of this section specifies the relationship of the MIB modules contained in this document to other standards, particularly to standards containing other MIB modules. Definitions imported from other MIB modules and other MIB modules that SHOULD be implemented in conjunction with the MIB module contained within this document are identified in this section.

6.1. Relationship to the SNMPv2-MIB

The 'system' group in the SNMPv2-MIB [RFC3418] is defined as being mandatory for all systems, and the objects apply to the entity as a whole. The 'system' group provides identification of the management entity and certain other system-wide data. The DYMO-MIB does not duplicate those objects.

6.2. MIB modules required for IMPORTS

The DYMO-MIB module IMPORTS objects from SNMPv2-SMI [RFC2578], SNMPv2-TC [RFC2579], SNMPv2-CONF [RFC2580], INET-ADDRESS-MIB [RFC4001] and IF-MIB [RFC2863].

7. Definitions

```
MANET-DYMO-MIB DEFINITIONS ::= BEGIN

IMPORTS

   MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
   Counter32, Integer32, Unsigned32, mib-2
      FROM SNMPv2-SMI                          -- [RFC2578]

   TEXTUAL-CONVENTION, RowStatus, TruthValue
      FROM SNMPv2-TC                           -- [RFC2579]

   MODULE-COMPLIANCE, OBJECT-GROUP,
   NOTIFICATION-GROUP
      FROM SNMPv2-CONF                         -- [RFC2580]

   InetAddress, InetAddressType,
   InetAddressPrefixLength
      FROM INET-ADDRESS-MIB                    -- [RFC4001]

   InterfaceIndexOrZero
      FROM IF-MIB                              -- [RFC2863]
   ;

manetDymoMIB MODULE-IDENTITY
   LAST-UPDATED "201101191200Z"  -- January 19, 2011
   ORGANIZATION "IETF MANET Working Group"
   CONTACT-INFO
      "WG E-Mail: manet@ietf.org
```

```
           WG Chairs: ian.chakeres@gmail.com
                      jmacker@nrl.navy.mil

           Editors:   Sean Harnedy
                      Booz Allen Hamilton
                      333 City Boulevard West
                      Orange, CA 92868
                      USA
                      +1 714 938-3898
                      harnedy_sean@bah.com

                      Robert G. Cole
                      US Army CERDEC
                      Space and Terrestrial Communications
                      328 Hopkins Road
                      Aberdeen Proving Ground, MD 21005
                      USA
                      +1 410 278-6779
                      robert.g.cole@us.army.mil

                      Ian D Chakeres
                      CenGen
                      9250 Bendix Road North
                      Columbia, Maryland  21045
                      USA
                      ian.chakeres@gmail.com"

      DESCRIPTION
         "This MIB module contains managed object definitions for
          the Dynamic MANET On-demand (DYMO) routing protocol as
          defined in: Chakeres,I., and C. Perkins, Dynamic MANET
          On-demand (DYMO) Routing, draft-ietf-manet-dymo-21,
          July 26, 2010.

          Copyright (C) The IETF Trust (2008). This version
          of this MIB module is part of RFC xxxx; see the RFC
          itself for full legal notices."

      -- Revision History
      REVISION    "201101191200Z"   -- January 19, 2011
      DESCRIPTION
         "Fifth draft of this MIB module published as
          draft-ietf-manet-dymo-mib-04.txt.
          Changes include:
          - Incorporated the DYMO ID by adding Instance
            Table.
          - Added dymoSetNotification for improved control
            of DYMO Notifications.
```

```
          - Updated various object names to be consistent
            with current draft-ietf-manet-dymo-21.
         "
      REVISION    "200910251200Z"   -- October 25, 2009
      DESCRIPTION
         "Fourth draft of this MIB module published as
          draft-ietf-manet-dymo-mib-03.txt.
          - Minor changes to textual material, including
            additions to the IMPORTS text.
          - Added DEFVAL clauses to all read-write
            configuration objects with defaults identified
            in the DYMO draft."
      REVISION    "200902241200Z"   -- February 24, 2009
      DESCRIPTION
         "Third draft of this MIB module published as
          draft-ietf-manet-dymo-mib-02.txt.
          - Minor changes to dymoInterfacesTable and
            dymoResponsibleAddrTable.
          - Added global dymoAdminStatus and interface
            specific dymoIfAdminStatus.
          - Imported InterfaceIndexOrZero type from
            IF-MIB."
      REVISION    "200811031200Z"   -- November 03, 2008
      DESCRIPTION
         "Second draft of this MIB module published as
          draft-ietf-manet-dymo-mib-01.txt. Minor changes to
          dymoInterfacesTable and dymoResponsibleAddrTable."
      REVISION    "200805141200Z"   -- May 14, 2008
      DESCRIPTION
         "Initial draft of this MIB module published as
          draft-ietf-manet-dymo-mib-00.txt."
      -- RFC-Editor assigns XXXX
      ::= { mib-2 999 }   -- to be assigned by IANA

   --
   -- TEXTUAL CONVENTIONs
   --

   Status ::= TEXTUAL-CONVENTION
       STATUS       current
       DESCRIPTION
          "An indication of the operability of a DYMO
          function or feature.  For example, the status
          of an interface: 'enabled' indicates that
          it is willing to communicate with other DYMO routers,
          and 'disabled' indicates that it is not."
       SYNTAX  INTEGER { enabled (1), disabled (2) }
```

```
   --
   -- Top-Level Object Identifier Assignments
   --

   dymoMIBNotifications OBJECT IDENTIFIER ::= { manetDymoMIB 0 }
   dymoMIBObjects       OBJECT IDENTIFIER ::= { manetDymoMIB 1 }
   dymoMIBConformance   OBJECT IDENTIFIER ::= { manetDymoMIB 2 }

   --
   -- dymoConfigurationGroup
   --
   --    This group contains the DYMO objects that configure specific
   --    options that determine the overall performance and operation
   --    of the routing protocol for the router device and its
   --    interfaces.
   --

   dymoConfigurationGroup  OBJECT IDENTIFIER ::= { dymoMIBObjects 1 }

   --
   -- DYMO Global Router Configuration Group
   --

  dymoRouterConfigGroup OBJECT IDENTIFIER ::= {dymoConfigurationGroup 1}

   dymoInstanceTable  OBJECT-TYPE
      SYNTAX      SEQUENCE OF DymoInstanceEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "The DYMO Instance Table describes the DYMO
          ...."
      REFERENCE
         "Dynamic MANET On-demand (DYMO) Routing, Chakeres,
          I., and C. Perkins, July 2010. The DID."
   ::= { dymoRouterConfigGroup 1 }

   dymoInstanceEntry OBJECT-TYPE
      SYNTAX      DymoInstanceEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "The DYMO instance entry describes one DYMO
          process as indexed by its DID."
      INDEX { dymoInstanceIndex }
   ::= { dymoInstanceTable 1 }

   DymoInstanceEntry ::=
```

```
      SEQUENCE {
         dymoInstanceIndex
            Integer32,
         dymoInstanceDid
            Integer32,
         dymoInstanceAdminStatus
            Status,
         dymoInstanceRowStatus
            RowStatus
         }

   dymoInstanceIndex  OBJECT-TYPE
      SYNTAX      Integer32 (0..255)
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "The instance index for this DYMO process."
      ::= { dymoInstanceEntry 1 }

   dymoInstanceDid OBJECT-TYPE
       SYNTAX       Integer32 (0..255)
       MAX-ACCESS   read-write
       STATUS       current
       DESCRIPTION
          "The DYMO ID of this instance of the
           DYMO process.
          "
   ::= { dymoInstanceEntry 2 }

   dymoInstanceAdminStatus OBJECT-TYPE
       SYNTAX       Status
       MAX-ACCESS   read-write
       STATUS       current
       DESCRIPTION
          "The administrative status of this DYMO
          process in the router.  Multiple processes are
          allowed.  The value 'enabled' denotes that the
          DYMO Process is active on at least one interface;
          'disabled' disables it on all interfaces.

          This object is persistent and when written
          the entity SHOULD save the change to non-volatile storage."
   ::= { dymoInstanceEntry 3 }

   dymoInstanceRowStatus  OBJECT-TYPE
      SYNTAX      RowStatus
      MAX-ACCESS  read-create
      STATUS      current
```

```
      DESCRIPTION
         "This object permits management of the table
          by facilitating actions such as row creation,
          construction, and destruction. The value of
          this object has no effect on whether other
          objects in this conceptual row can be
          modified."
   ::= { dymoInstanceEntry 4 }

   dymoMaxHopLimit  OBJECT-TYPE
      SYNTAX      Unsigned32 (0..255)
      UNITS       "hops"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "The maximum number of hops. The suggested value
          default is 10 hops. This is the DYMO MSG_HOPLIMIT
          parameter value."
      REFERENCE
         "Dynamic MANET On-demand (DYMO) Routing, Chakeres,
          I., and C. Perkins, July 2010. Table 2 Suggested
          Parameter Values."
      DEFVAL { 10 }
   ::= { dymoRouterConfigGroup 2 }

   dymoRouteTimeout  OBJECT-TYPE
      SYNTAX      Unsigned32 (1..65535)
      UNITS       "milliseconds"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "The route timeout value. The suggested default
          value is 5000 milliseconds. This is the
          DYMO ROUTE_TIMEOUT parameter value."
      REFERENCE
         "Dynamic MANET On-demand (DYMO) Routing, Chakeres,
          I., and C. Perkins, July 2010. Table 2 Suggested
          Parameter Values."
      DEFVAL { 5000 }
   ::= { dymoRouterConfigGroup 3 }

   dymoRouteAgeMinTimeout  OBJECT-TYPE
      SYNTAX      Unsigned32 (1..65535)
      UNITS       "milliseconds"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "The minimum route age timeout value. The
```

```
          suggested default value is 1000 milliseconds.
          This is the DYMO ROUTE_AGE_MIN_TIMEOUT parameter
          value."
      REFERENCE
         "Dynamic MANET On-demand (DYMO) Routing, Chakeres,
          I., and C. Perkins, July 2010. Table 2 Suggested
          Parameter Values."
      DEFVAL { 1000 }
   ::= { dymoRouterConfigGroup 4 }

   dymoRouteSeqnumAgeMaxTimeout  OBJECT-TYPE
      SYNTAX      Unsigned32 (1..65535)
      UNITS       "milliseconds"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "The maximum route age timeout value. The
          suggested default value is 60,000 milliseconds.
          This is the DYMO ROUTE_SEQNUM_AGE_MAX_TIMEOUT
          parameter value."
      REFERENCE
         "Dynamic MANET On-demand (DYMO) Routing, Chakeres,
          I., and C. Perkins, July 2010. Table 2 Suggested
          Parameter Values."
      DEFVAL { 60000 }
   ::= { dymoRouterConfigGroup 5 }

   dymoRouteUsedTimeout  OBJECT-TYPE
      SYNTAX      Unsigned32 (1..65535)
      UNITS       "milliseconds"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "The route used timeout value. The
          suggested default value is to set this
          to the dymoRouteTimeout object value
          (whose default is 5000 milliseconds). This
          is the DYMO ROUTE_USED_TIMEOUT parameter
          value."
      REFERENCE
         "Dynamic MANET On-demand (DYMO) Routing, Chakeres,
          I., and C. Perkins, July 2010. Table 2 Suggested
          Parameter Values."
      DEFVAL { 5000 }
   ::= { dymoRouterConfigGroup 6 }

   dymoRouteDeleteTimeout  OBJECT-TYPE
      SYNTAX      Unsigned32 (1..65535)
```

```
      UNITS       "milliseconds"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "The route delete timeout value. The
          suggested default value is 2 * dymoRouteTimeout
          value (which is equal to 10000 milliseconds
          if using the default value for the
          dymoRouteTimeout value). This is the
          DYMO ROUTE_DELETE_TIMEOUT parameter value."
      REFERENCE
         "Dynamic MANET On-demand (DYMO) Routing, Chakeres,
          I., and C. Perkins, July 2010. Table 2 Suggested
          Parameter Values."
      DEFVAL { 10000 }
   ::= { dymoRouterConfigGroup 7 }

   dymoRouteRreqWaitTime  OBJECT-TYPE
      SYNTAX      Unsigned32 (1..65535)
      UNITS       "milliseconds"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "The Route Request wait time. The suggested default
          value is 2000 milliseconds. This is the DYMO
          ROUTE_RREQ_WAIT_TIME parameter value."
      REFERENCE
         "Dynamic MANET On-demand (DYMO) Routing, Chakeres,
          I., and C. Perkins, July 2010. Table 2 Suggested
          Parameter Values."
      DEFVAL { 2000 }
   ::= { dymoRouterConfigGroup 8 }

   dymoDiscoveryAttemptsMax  OBJECT-TYPE
      SYNTAX      Unsigned32 (1..16)
      UNITS       "attempts"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "The number of Route Request retry attempts. The
          suggested default value is 3. This is the
          DYMO DISCOVERY_ATTEMPTS_MAX parameter value."
      REFERENCE
         "Dynamic MANET On-demand (DYMO) Routing, Chakeres,
          I., and C. Perkins, July 2010. Table 2 Suggested
          Parameter Values."
      DEFVAL { 3 }
   ::= { dymoRouterConfigGroup 9 }
```

```
   dymoUnicastMsgSentTimeout  OBJECT-TYPE
      SYNTAX      Unsigned32 (1..65535)
      UNITS       "milliseconds"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "The message sent timeout value for unicast packets.
          The suggested default value is 1000 milliseconds.
          This is the DYMO UNICAST_MESSAGE_SENT_TIMEOUT
          parameter value."
      REFERENCE
         "Dynamic MANET On-demand (DYMO) Routing, Chakeres,
          I., and C. Perkins, July 2010. Table 2 Suggested
          Parameter Values."
      DEFVAL { 1000 }
   ::= { dymoRouterConfigGroup 10 }



   --
   -- DYMO Interfaces Configuration Table
   --

   dymoInterfaceTable  OBJECT-TYPE
      SYNTAX      SEQUENCE OF DymoInterfaceEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "The DYMO Interface Table describes the DYMO
          interfaces that are participating in the
          DYMO routing protocol. The ifIndex is from
          the interfaces group defined in the Interfaces
          Group MIB."
      REFERENCE
         "RFC 2863 - The Interfaces Group MIB, McCloghrie,
          K., and F. Kastenholtz, June 2000."
   ::= { dymoConfigurationGroup 2 }

   dymoInterfaceEntry OBJECT-TYPE
      SYNTAX      DymoInterfaceEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "The DYMO interface entry describes one DYMO
          interface as indexed by its ifIndex."
      INDEX { dymoIfIndex }
   ::= { dymoInterfaceTable 1 }
```

```
   DymoInterfaceEntry ::=
      SEQUENCE {
         dymoIfIndex
            InterfaceIndexOrZero,
         dymoIfAdminStatus
            Status,
         dymoIfRowStatus
            RowStatus
         }

   dymoIfIndex  OBJECT-TYPE
      SYNTAX      InterfaceIndexOrZero
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "The ifIndex for this DYMO interface."
      ::= { dymoInterfaceEntry 1 }

   dymoIfAdminStatus OBJECT-TYPE
      SYNTAX      Status
      MAX-ACCESS  read-create
      STATUS      current
      DESCRIPTION
          "The DYMO interface's administrative status.
          The value 'enabled' denotes that the interface
          is running the DYMO routing protocol.
          The value 'disabled' denotes that the interface is
          external to DYMO."
      ::= { dymoInterfaceEntry 2 }

   dymoIfRowStatus  OBJECT-TYPE
      SYNTAX      RowStatus
      MAX-ACCESS  read-create
      STATUS      current
      DESCRIPTION
         "This object permits management of the table
          by facilitating actions such as row creation,
          construction, and destruction. The value of
          this object has no effect on whether other
          objects in this conceptual row can be
          modified."
   ::= { dymoInterfaceEntry 3 }


   --
   -- DYMO Responsible Address Table
   --
```

```
   dymoResponsibleAddrTable OBJECT-TYPE
      SYNTAX       SEQUENCE OF DymoResponsibleAddrEntry
      MAX-ACCESS   not-accessible
      STATUS       current
      DESCRIPTION
         "The DYMO Responsible Address Table is a
          list of IP address prefixes, and their
          associated prefix length for which the
          DYMO router is responsible."
      REFERENCE
         "Dynamic MANET On-demand (DYMO) Routing, Chakeres,
          I., and C. Perkins, July 2010. Table 3 Important
          Settings."
   ::= { dymoConfigurationGroup 3 }

   dymoResponsibleAddrEntry  OBJECT-TYPE
      SYNTAX      DymoResponsibleAddrEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "A single host address range. Information
          in this table is persistent and when this object
          is written, the entity SHOULD save the change to
          non-volatile storage."
      REFERENCE
         "Dynamic MANET On-demand (DYMO) Routing, Chakeres,
          I., and C. Perkins, July 2010. Table 3 Important
          Settings."
      INDEX { dymoResponsibleAddrIndex }
   ::= { dymoResponsibleAddrTable 1 }

   DymoResponsibleAddrEntry ::=
      SEQUENCE {
         dymoResponsibleAddrIndex
            Unsigned32,
         dymoResponsibleAddrType
            InetAddressType,
         dymoResponsibleAddr
            InetAddress,
         dymoResponsibleAddrPrefixLen
            InetAddressPrefixLength,
         dymoResponsibleAddrRowStatus
            RowStatus
         }

   dymoResponsibleAddrIndex  OBJECT-TYPE
      SYNTAX      Unsigned32
      MAX-ACCESS  not-accessible
```

```
      STATUS      current
      DESCRIPTION
         "This object is the index into this table."
   ::= { dymoResponsibleAddrEntry 1 }

   dymoResponsibleAddrType  OBJECT-TYPE
      SYNTAX      InetAddressType
      MAX-ACCESS  read-create
      STATUS      current
      DESCRIPTION
         "The type of the dymoResponsibleAddr, as defined
          in the InetAddress MIB [RFC 4001]."
      REFERENCE
         "Dynamic MANET On-demand (DYMO) Routing, Chakeres,
          I., and C. Perkins, July 2010. Table 3 Important
          Settings."
   ::= { dymoResponsibleAddrEntry 2 }

   dymoResponsibleAddr  OBJECT-TYPE
      SYNTAX      InetAddress
      MAX-ACCESS  read-create
      STATUS      current
      DESCRIPTION
         "The destination IP address of this route. The type
          of this address is determined by the value of the
          dymoResponsibleAddrType object."
      REFERENCE
         "Dynamic MANET On-demand (DYMO) Routing, Chakeres,
          I., and C. Perkins, July 2010. Table 3 Important
          Settings."
   ::= { dymoResponsibleAddrEntry 3 }

   dymoResponsibleAddrPrefixLen  OBJECT-TYPE
      SYNTAX      InetAddressPrefixLength
      MAX-ACCESS  read-create
      STATUS      current
      DESCRIPTION
         "Indicates the number of leading one bits that form the
          mask to be logical-AND'd with the destination address
          before being compared to the value in the dymoResonsibleAddr
          field."
      REFERENCE
         "Dynamic MANET On-demand (DYMO) Routing, Chakeres,
          I., and C. Perkins, July 2010. Table 3 Important
          Settings."
   ::= { dymoResponsibleAddrEntry 4 }

   dymoResponsibleAddrRowStatus  OBJECT-TYPE
```

```
      SYNTAX      RowStatus
      MAX-ACCESS  read-create
      STATUS      current
      DESCRIPTION
         "This object permits management of the table
          by facilitating actions such as row creation,
          construction, and destruction. The value of
          this object has no effect on whether other
          objects in this conceptual row can be
          modified."
   ::= { dymoResponsibleAddrEntry 5 }


   --
   -- dymoStateGroup
   --
   --    Contains information describing the current state of the DYMO
   --    process such as the DYMO routing table.
   --

   dymoStateGroup  OBJECT IDENTIFIER ::= { dymoMIBObjects 2 }

   dymoCurrentSeqNum  OBJECT-TYPE
      SYNTAX      Unsigned32 (1..65535)
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The current DYMO sequence number. The DYMO sequence
          numbers allow nodes to judge the freshness of routing
          information and ensures loop freedom. If the sequence
          number has been assigned to be the largest possible
          number representable as a 16-bit unsigned integer
          (i.e., 65,535), then the sequence number is set to
          256 when incremented.  Setting the sequence number
          to 256 allows other nodes to detect that the number
          has rolled over and the node has not lost its sequence
          number (e.g., via reboot)."
   ::= { dymoStateGroup 1 }

   --
   -- DYMO Routing Table
   --

   dymoRoutingTable  OBJECT-TYPE
      SYNTAX       SEQUENCE OF DymoRoutingEntry
      MAX-ACCESS   not-accessible
      STATUS       current
      DESCRIPTION
```

```
         "The DYMO Routing Table describes the
          current routing information learned
          via DYMO control messages."
      REFERENCE
         "Dynamic MANET On-demand (DYMO) Routing, Chakeres,
          I., and C. Perkins, July 2010. Table 2 Suggested
          Parameter Values."
   ::= { dymoStateGroup 2 }

   dymoRoutingEntry  OBJECT-TYPE
      SYNTAX       DymoRoutingEntry
      MAX-ACCESS   not-accessible
      STATUS       current
      DESCRIPTION
         "The DYMO routing entry contains a
          piece of routing information for a
          particular set of addresses."
      INDEX { dymoRoutingIpAddrType,
              dymoRoutingIpAddr,
              dymoRoutingPrefixLen }
   ::= { dymoRoutingTable 1 }

   DymoRoutingEntry ::=
      SEQUENCE {
         dymoRoutingIpAddrType
            InetAddressType,
         dymoRoutingIpAddr
            InetAddress,
         dymoRoutingPrefixLen
            InetAddressPrefixLength,
         dymoRoutingSeqNum
            Unsigned32,
         dymoRoutingNextHopIpAddrType
            InetAddressType,
         dymoRoutingNextHopIpAddress
            InetAddress,
         dymoRoutingNextHopInterface
            InterfaceIndexOrZero,
         dymoRoutingForwardingFlag
            TruthValue,
         dymoRoutingBrokenFlag
            TruthValue,
         dymoRoutingDist
            Unsigned32
         }

   dymoRoutingIpAddrType  OBJECT-TYPE
      SYNTAX      InetAddressType
```

```
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "The routing table address IP address type."
      REFERENCE
         "Dynamic MANET On-demand (DYMO) Routing, Chakeres,
          I., and C. Perkins, July 2010. Table 3 Important
          Settings."
   ::= { dymoRoutingEntry 1 }

   dymoRoutingIpAddr  OBJECT-TYPE
      SYNTAX      InetAddress
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "The routing table Inet IPv4 or IPv6 address."
      REFERENCE
         "Dynamic MANET On-demand (DYMO) Routing, Chakeres,
          I., and C. Perkins, July 2010. Table 3 Important
          Settings."
   ::= { dymoRoutingEntry 2 }

   dymoRoutingPrefixLen  OBJECT-TYPE
      SYNTAX      InetAddressPrefixLength
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "The prefix length. This is a decimal value that
          indicates the number of contiguous, higher-order
          bits of the address that make up the network
          portion of the address."
      REFERENCE
         "Dynamic MANET On-demand (DYMO) Routing, Chakeres,
          I., and C. Perkins, July 2010. Table 3 Important
          Settings."
   ::= { dymoRoutingEntry 3 }

   dymoRoutingSeqNum  OBJECT-TYPE
      SYNTAX       Unsigned32 (1..65535)
      MAX-ACCESS   read-only
      STATUS       current
      DESCRIPTION
         "The interface sequence number. This
          is the DYMO SeqNum associated with this
          routing information."
   ::= { dymoRoutingEntry 4 }

   dymoRoutingNextHopIpAddrType OBJECT-TYPE
```

```
      SYNTAX      InetAddressType
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The IP address type of the next hop."
      ::= { dymoRoutingEntry 5 }

   dymoRoutingNextHopIpAddress OBJECT-TYPE
      SYNTAX       InetAddress
      MAX-ACCESS   read-only
      STATUS       current
      DESCRIPTION
         "The IP address of the next hop."
      ::= { dymoRoutingEntry 6 }

   dymoRoutingNextHopInterface OBJECT-TYPE
      SYNTAX       InterfaceIndexOrZero
      MAX-ACCESS   read-only
      STATUS       current
      DESCRIPTION
         "The interface ifIndex for sending
          packets toward the destination route
          address."
      ::= { dymoRoutingEntry 7 }

   dymoRoutingForwardingFlag OBJECT-TYPE
      SYNTAX       TruthValue
      MAX-ACCESS   read-only
      STATUS       current
      DESCRIPTION
         "The Forwarding Flag indicates whether
          this route can be used for forwarding
          data packets. A value 'true(1)'
          indicates that this route is being used
          for forwarding of data packets, while
          a value 'false(2)' indicates that it is
          not being used for forwarding."
      ::= { dymoRoutingEntry 8 }

   dymoRoutingBrokenFlag OBJECT-TYPE
      SYNTAX       TruthValue
      MAX-ACCESS   read-only
      STATUS       current
      DESCRIPTION
         "The Broken Flag indicates whether
          this Route is broken.  This flag is set
          if the next-hop becomes unreachable or
          in response to processing a RERR. A value
```

```
          'true(1)' indicates that this route is
          broken, while a value 'false(2)'
          indicates that it is not broken."
      ::= { dymoRoutingEntry 9 }

   dymoRoutingDist OBJECT-TYPE
      SYNTAX       Unsigned32 (0..65535)
      UNITS        "hops"
      MAX-ACCESS   read-only
      STATUS       current
      DESCRIPTION
         "The distance to the destination address's
          DYMO router. This is a metric of the
          distance a message or piece of information
          has traversed. The minimum value of distance
          is the number of IP hops traversed. The
          maximum value is 65,535.

          This parameter is an optional field in the
          DYMO routing table.  If the DYMO Route.Dist
          is not supported by this device, then this
          object should be set to '0'."
      REFERENCE
         "Dynamic MANET On-demand (DYMO) Routing,
          Chakeres, I., and C. Perkins, April
          2008. Section 3 Terminology."
      ::= { dymoRoutingEntry 10 }




   --
   -- DYMO Performance Group (Performance Management)
   --
   --    Contains objects which help to characterize the
   --    performance of the DYMO process, typically statistics
   --    counters. There are two types of DYMO statistics:
   --    global counters and per interface counters.
   --

   dymoPerformanceGroup  OBJECT IDENTIFIER ::= { dymoMIBObjects 3 }

   dymoGlobalPerfGroup  OBJECT IDENTIFIER ::= { dymoPerformanceGroup 1 }

   dymoRreqOriginated  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
```

```
      STATUS      current
      DESCRIPTION
         "A counter of the number of
          RREQ messages that this DYMO
          device has initiated."
   ::= { dymoGlobalPerfGroup 1 }

   dymoRreqForwarded  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter of the number of
          RREQ messages that this DYMO
          device has forwarded, i.e., this
          device neither originated or
          terminated the RREQ message."
   ::= { dymoGlobalPerfGroup 2 }

   dymoRreqReceived  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter of the number of
          RREQ messages that this DYMO
          device has received as the
          target of the message."
   ::= { dymoGlobalPerfGroup 3 }

   dymoRrepOriginated  OBJECT-TYPE
      SYNTAX        Counter32
      MAX-ACCESS    read-only
      STATUS        current
      DESCRIPTION
         "A counter of the number of
          RREP messages that this DYMO
          device has initiated."
   ::= { dymoGlobalPerfGroup 4 }

   dymoRrepForwarded  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter of the number of
          RREP messages that this DYMO
          device has forwarded, i.e, this
```

```
          device neither originated or
          terminated the RREP message."
   ::= { dymoGlobalPerfGroup 5 }

   dymoRrepReceived   OBJECT-TYPE
      SYNTAX        Counter32
      MAX-ACCESS    read-only
      STATUS        current
      DESCRIPTION
         "A counter of the number of
          RREP messages that this DYMO
          device has received as the
          target of the message."
   ::= { dymoGlobalPerfGroup 6 }

   dymoRrerOriginated  OBJECT-TYPE
      SYNTAX        Counter32
      MAX-ACCESS    read-only
      STATUS        current
      DESCRIPTION
         "A counter of the number of
          RRER messages that this DYMO
          device has initiated."
   ::= { dymoGlobalPerfGroup 7 }

   dymoRrerForwarded  OBJECT-TYPE
      SYNTAX        Counter32
      MAX-ACCESS    read-only
      STATUS        current
      DESCRIPTION
         "A counter of the number of
          RRER messages that this DYMO
          device has forwarded, i.e., this
          device neither originated or
          terminated the RRER message."
   ::= { dymoGlobalPerfGroup 8 }

   dymoRrerReceived  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter of the number of
          RRER messages that this DYMO
          device has received as the
          target of the message."
   ::= { dymoGlobalPerfGroup 9 }
```

```
   --
   -- Per DYMO Interface Performance Table
   --

   dymoInterfacePerfGroup OBJECT IDENTIFIER ::= {dymoPerformanceGroup 2}

   dymoInterfacePerfTable OBJECT-TYPE
      SYNTAX       SEQUENCE OF DymoInterfacePerfEntry
      MAX-ACCESS   not-accessible
      STATUS       current
      DESCRIPTION
         "The DYMO Interface Performance Table
          describes the DYMO statistics per
          interface."
   ::= { dymoInterfacePerfGroup 1 }

   dymoInterfacePerfEntry OBJECT-TYPE
      SYNTAX       DymoInterfacePerfEntry
      MAX-ACCESS   not-accessible
      STATUS       current
      DESCRIPTION
         "The DYMO Interface Performance entry
          describes the statistics for a particular
          DYMO interface."
      INDEX { dymoIfPerfIfIndex }
   ::= { dymoInterfacePerfTable 1 }

   DymoInterfacePerfEntry ::=
      SEQUENCE {
         dymoIfPerfIfIndex
            InterfaceIndexOrZero,
         dymoIfRreqOriginated
            Counter32,
         dymoIfRreqForwarded
            Counter32,
         dymoIfRreqReceived
            Counter32,
         dymoIfRrepOriginated
            Counter32,
         dymoIfRrepForwarded
            Counter32,
         dymoIfRrepReceived
            Counter32,
         dymoIfRrerOriginated
            Counter32,
         dymoIfRrerForwarded
            Counter32,
         dymoIfRrerReceived
```

```
            Counter32
            }

   dymoIfPerfIfIndex  OBJECT-TYPE
      SYNTAX      InterfaceIndexOrZero
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "The ifIndex for this DYMO interface
          that is collecting this set of
          performance management statistics."
   ::= { dymoInterfacePerfEntry 1 }

   dymoIfRreqOriginated  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter of the number of
          RREQ messages that this DYMO
          interface has initiated."
   ::= { dymoInterfacePerfEntry 2 }

   dymoIfRreqForwarded  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter of the number of
          RREQ messages that this DYMO
          interface has forwarded, i.e., this
          interface neither originated nor
          terminated the RREQ message."
   ::= { dymoInterfacePerfEntry 3 }

   dymoIfRreqReceived  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter of the number of
          RREQ messages that this DYMO
          interface has received as the
          target of the message."
   ::= { dymoInterfacePerfEntry 4 }

   dymoIfRrepOriginated  OBJECT-TYPE
      SYNTAX        Counter32
```

```
      MAX-ACCESS    read-only
      STATUS        current
      DESCRIPTION
         "A counter of the number of
          RREP messages that this DYMO
          interface has initiated."
   ::= { dymoInterfacePerfEntry 5 }

   dymoIfRrepForwarded  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter of the number of
          RREP messages that this DYMO
          interface has forwarded, i.e., this
          interface neither originated nor
          terminated the RREP message."
   ::= { dymoInterfacePerfEntry 6 }

   dymoIfRrepReceived   OBJECT-TYPE
      SYNTAX        Counter32
      MAX-ACCESS    read-only
      STATUS        current
      DESCRIPTION
         "A counter of the number of
          RREP messages that this DYMO
          interface has received as the
          target of the message."
   ::= { dymoInterfacePerfEntry 7 }

   dymoIfRrerOriginated  OBJECT-TYPE
      SYNTAX        Counter32
      MAX-ACCESS    read-only
      STATUS        current
      DESCRIPTION
         "A counter of the number of
          RRER messages that this DYMO
          interface has initiated."
   ::= { dymoInterfacePerfEntry 8 }

   dymoIfRrerForwarded  OBJECT-TYPE
      SYNTAX        Counter32
      MAX-ACCESS    read-only
      STATUS        current
      DESCRIPTION
         "A counter of the number of
          RRER messages that this DYMO
```

```
          interface has forwarded, i.e., this
          interface neither originated nor
          terminated the RRER message."
   ::= { dymoInterfacePerfEntry 9 }

   dymoIfRrerReceived  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter of the number of
          RRER messages that this DYMO
          interface has received as the
          target of the message."
   ::= { dymoInterfacePerfEntry 10 }




   --
   -- Notifications
   --

   dymoMIBNotifControl OBJECT IDENTIFIER ::= { dymoMIBNotifications 1 }
   dymoMIBNotifObjects OBJECT IDENTIFIER ::= { dymoMIBNotifications 2 }

   -- dymoMIBNotifControl

   dymoSetNotification OBJECT-TYPE
          SYNTAX       OCTET STRING (SIZE(4))
          MAX-ACCESS   read-write
          STATUS       current
          DESCRIPTION
             "A 4-octet string serving as a bit map for
             the notification events defined by the DYMO
             notifications. This object is used to enable
             and disable specific DYMO notifications where
             a 1 in the bit field represents enabled. The
             right-most bit (least significant) represents
             notification 0.

             This object is persistent and when written
             the entity SHOULD save the change to
             non-volatile storage.
             "
           ::= { dymoMIBNotifControl 1 }
```

```
   -- dymoMIBNotifObjects

   dymoInstanceAdminStatusChange  NOTIFICATION-TYPE
      OBJECTS      { dymoInstanceAdminStatus,
                     dymoInstanceDid
                   }
      STATUS       current
      DESCRIPTION
         "This notification is generated when the
          administrative status of a DYMO process changes."
   ::= { dymoMIBNotifObjects 1 }

   dymoInterfaceAdminStatusChange  NOTIFICATION-TYPE
      OBJECTS      { dymoIfAdminStatus }
      STATUS       current
      DESCRIPTION
         "This notification is generated when the
          administrative status of a DYMO interface changes."
   ::= { dymoMIBNotifObjects 2 }

   dymoResponsibleAddrEntryChange  NOTIFICATION-TYPE
      OBJECTS      { dymoResponsibleAddrRowStatus }
      STATUS       current
      DESCRIPTION
         "This notification is generated when the status
          of an entry in the DYMO Responsible Address
          Table changes. This includes the creation or
          deletion of a row."
   ::= { dymoMIBNotifObjects 3 }

   --
   -- Compliance Statements
   --

   dymoCompliances  OBJECT IDENTIFIER ::= { dymoMIBConformance 1 }
   dymoMIBGroups    OBJECT IDENTIFIER ::= { dymoMIBConformance 2 }

   dymoBasicCompliance  MODULE-COMPLIANCE
      STATUS current
      DESCRIPTION "The basic implementation requirements for
                   managed network entities that implement
                   the DYMO routing protocol."
      MODULE  -- this module
      MANDATORY-GROUPS { dymoConfigObjectsGroup }
   ::= { dymoCompliances 1 }

   dymoFullCompliance MODULE-COMPLIANCE
      STATUS current
```

```
      DESCRIPTION "The full implementation requirements for managed
                   network entities that implement the DYMO routing
                   protocol."
      MODULE  -- this module
      MANDATORY-GROUPS { dymoConfigObjectsGroup,
                         dymoStateObjectsGroup,
                         dymoPerfObjectsGroup,
                         dymoNotifObjectsGroup,
                         dymoNotificationGroup }
   ::= { dymoCompliances 2 }

   --
   -- Units of Conformance
   --

   dymoConfigObjectsGroup OBJECT-GROUP
      OBJECTS {
              dymoInstanceAdminStatus,
              dymoInstanceDid,
              dymoInstanceRowStatus,
              dymoMaxHopLimit,
              dymoRouteTimeout,
              dymoRouteAgeMinTimeout,
              dymoRouteSeqnumAgeMaxTimeout,
              dymoRouteUsedTimeout,
              dymoRouteDeleteTimeout,
              dymoRouteRreqWaitTime,
              dymoDiscoveryAttemptsMax,
              dymoUnicastMsgSentTimeout,
              dymoIfAdminStatus,
              dymoIfRowStatus,
              dymoResponsibleAddrType,
              dymoResponsibleAddr,
              dymoResponsibleAddrPrefixLen,
              dymoResponsibleAddrRowStatus
      }
      STATUS  current
      DESCRIPTION
         "Set of DYMO configuration objects implemented
          in this module."
   ::= { dymoMIBGroups 1 }

   dymoStateObjectsGroup  OBJECT-GROUP
      OBJECTS {
              dymoCurrentSeqNum,
              dymoRoutingSeqNum,
              dymoRoutingNextHopIpAddrType,
              dymoRoutingNextHopIpAddress,
```

```
              dymoRoutingNextHopInterface,
              dymoRoutingForwardingFlag,
              dymoRoutingBrokenFlag,
              dymoRoutingDist
      }
      STATUS  current
      DESCRIPTION
         "Set of DYMO state objects implemented
          in this module."
   ::= { dymoMIBGroups 2 }

  dymoPerfObjectsGroup  OBJECT-GROUP
      OBJECTS {
              dymoRreqOriginated,
              dymoRreqForwarded,
              dymoRreqReceived,
              dymoRrepOriginated,
              dymoRrepForwarded,
              dymoRrepReceived,
              dymoRrerOriginated,
              dymoRrerForwarded,
              dymoRrerReceived,
              dymoIfRreqOriginated,
              dymoIfRreqForwarded,
              dymoIfRreqReceived,
              dymoIfRrepOriginated,
              dymoIfRrepForwarded,
              dymoIfRrepReceived,
              dymoIfRrerOriginated,
              dymoIfRrerForwarded,
              dymoIfRrerReceived
      }
      STATUS  current
      DESCRIPTION
         "Set of DYMO statistic objects implemented
          in this module for performance management."
   ::= { dymoMIBGroups 3 }

   dymoNotifObjectsGroup OBJECT-GROUP
      OBJECTS {
         dymoSetNotification
      }
      STATUS  current
      DESCRIPTION
         "Set of DYMO notifications objects implemented
          in this module."
   ::= { dymoMIBGroups 4 }
```

```
   dymoNotificationGroup NOTIFICATION-GROUP
      NOTIFICATIONS {
         dymoInstanceAdminStatusChange,
         dymoInterfaceAdminStatusChange,
         dymoResponsibleAddrEntryChange
      }
      STATUS  current
      DESCRIPTION
         "Set of DYMO notifications implemented in this
          module."
   ::= { dymoMIBGroups 5 }

   END
```

8. Security Considerations

[TODO] Each specification that defines one or more MIB modules MUST contain a section that discusses security considerations relevant to those modules. This section MUST be patterned after the latest approved template (available at http://www.ops.ietf.org/mib-security.html). Remember that the objective is not to blindly copy text from the template, but rather to think and evaluate the risks/vulnerabilities and then state/ document the result of this evaluation.

[TODO] if you have any read-write and/or read-create objects, please include the following boilerplate paragraph.

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 can have a negative effect on network operations. These are the tables and objects and their sensitivity/vulnerability:

- o [TODO] writable MIB objects that could be especially disruptive if abused MUST be explicitly listed by name and the associated security risks MUST be spelled out; RFC 2669 has a very good example.

- o [TODO] list the writable tables and objects and state why they are sensitive.

[TODO] else if there are no read-write objects in your MIB module, use the following boilerplate paragraph.

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.

[TODO] if you have any sensitive readable objects, please include the following boilerplate paragraph.

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:

o [TODO] you must explicitly list by name any readable objects that are sensitive or vulnerable and the associated security risks MUST be spelled out (for instance, if they might reveal customer information or violate personal privacy laws such as those of the European Union if exposed to unauthorized parties)

o [TODO] list the tables and objects and state why they are sensitive.

[TODO] discuss what security the protocol used to carry the information should have. The following three boilerplate paragraphs should not be changed without very good reason. Changes will almost certainly require justification during IESG review.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPSec), even then, there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

It is RECOMMENDED that implementers consider the security features as provided by the SNMPv3 framework (see [RFC3410], section 8), including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

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.

9. IANA Considerations

[TODO] In order to comply with IESG policy as set forth in http://www.ietf.org/ID-Checklist.html, every Internet-Draft that is submitted to the IESG for publication MUST contain an IANA Considerations section. The requirements for this section vary depending what actions are required of the IANA. see RFC4181 section 3.5 for more information on writing an IANA clause for a MIB module document.

[TODO] select an option and provide the necessary details.

Option #1:

The MIB module in this document uses the following IANA-assigned OBJECT IDENTIFIER values recorded in the SMI Numbers registry:

```
Descriptor        OBJECT IDENTIFIER value
----------        -----------------------

sampleMIB  { mib-2 XXX }
```

Option #2:

Editor's Note (to be removed prior to publication): the IANA is requested to assign a value for "XXX" under the 'mib-2' subtree and to record the assignment in the SMI Numbers registry. When the assignment has been made, the RFC Editor is asked to replace "XXX" (here and in the MIB module) with the assigned value and to remove this note.

Note well: prior to official assignment by the IANA, a draft document MUST use place-holders (such as "XXX" above) rather than actual numbers. See RFC4181 Section 4.5 for an example of how this is done in a draft MIB module.

Option #3:

This memo includes no request to IANA.

10. Contributors

This MIB document uses the template authored by D. Harrington which is based on contributions from the MIB Doctors, especially Juergen Schoenwaelder, Dave Perkins, C.M.Heard and Randy Presuhn.

11. Acknowledgements

12. References

12.1. Normative References

[RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.

[RFC3418] Presuhn, R., "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3418, December 2002.

[RFC4001] Daniele, M., Haberman, B., Routhier, S., and J. Schoenwaelder, "Textual Conventions for Internet Network Addresses", RFC 4001, February 2005.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

[I-D.ietf-manet-dymo] Chakeres, I. and C. Perkins, "Dynamic MANET On-demand (DYMO) Routing", draft-ietf-manet-dymo-21 (work in progress), July 2010.

12.2. Informative References

[RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

Appendix A. Change Log

This section identifies the changes that have been made from draft-ietf-manet-dymo-mib-00 .

These changes were made from draft-ietf-manet-dymo-mib-00 to draft-ietf-manet-dymo-mib-01.

1. Only minor changes of a typographic nature, e.g., read-only to read-write on MAX_ACCESS clauses of a few configuration objects.

These changes were made from draft-ietf-manet-dymo-mib-01 to draft-ietf-manet-dymo-mib-02.

1. Added the ForwardingFlag and BrokenFlag objects to the DYMO Routing Table.

2. Added the TruthValue Textual Convention to handle the new Routing Table objects.

3. Added the DYMO device management model to the introductory sections of this draft.

4. General clean up of the introductory sections of this draft.

These changes were made from draft-ietf-manet-dymo-mib-02 to draft-ietf-manet-dymo-mib-03.

1. Minor changes to the textual material and added to the IMPORTS text in the introductory material.

2. Added DEFVAL clauses to all read-write configuration objects having default values identified in the DYMO specification.

These changes were made from draft-ietf-manet-dymo-mib-03 to draft-ietf-manet-dymo-mib-04.

1. Incorporated the DID into the Configuration Group by changing the dymoAdminStatus object to an Instance Table. This allows for the presence of multiple DYMO processes concurrent on the same router.

2. Added the dymoNotifObjectsGroup and its dymoSetNotifications object to allow for individual control of the DYMO Notifications. Updated the Conformance sections accordingly.

3. Renamed several of the Configuration Objects to be consistent with the naming within the current draft-ietf-manet-dymo-21.

Appendix B. Open Issues

This section contains the set of open issues related to the development and design of the DYMO-MIB. This section will not be present in the final version of the MIB and will be removed once all the open issues have been resolved.

1. Work on the Security Section. This MIB does have settable objects, but not sensitive objects (true?).

2. Work on the relationship to other MIBs, IF-MIB, NHDP-MIB.

3. Cleanup all the [TODOs] from the MIB template.

Appendix C.

```
**************************************************************
* Note to the RFC Editor (to be removed prior to publication) *
*                                                             *
* 1) The reference to RFCXXXX within the DESCRIPTION clauses  *
* of the MIB module point to this draft and are to be         *
* assigned by the RFC Editor.                                 *
*                                                             *
* 2) The reference to RFCXXX2 throughout this document point  *
* to the current draft-ietf-manet-dymo-xx.txt.  This          *
* need to be replaced with the XXX RFC number.                *
*                                                             *
**************************************************************
```

Authors' Addresses

Sean Harnedy
Booz Allen Hamilton
333 City Boulevard West
Orange, California 92868
USA

Phone: +1 714 938-3898
EMail: harnedy_sean@bah.com

Robert G. Cole
US Army CERDEC
328 Hopkins Road, Bldg 245
Aberdeen Proving Ground, Maryland 21005
USA

Phone: +1 410 278 6779
EMail: robert.g.cole@us.army.mil
URI: http://www.cs.jhu.edu/~rgcole/

Ian D Chakeres
CenGen
9250 Bendix Road North
Columbia, Maryland 21045
USA

EMail: ian.chakeres@gmail.com
URI: http://www.ianchak.com/

Internet Engineering Task Force U. Herberg
Internet-Draft LIX, Ecole Polytechnique
Intended status: Standards Track R. Cole
Expires: July 7, 2011 US Army CERDEC
I. Chakeres
CenGen
January 3, 2011

Definition of Managed Objects for the Neighborhood Discovery Protocol
draft-ietf-manet-nhdp-mib-07

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 objects for configuring parameters of the Neighborhood Discovery Protocol (NHDP) process on a router. The MIB defined in this memo, denoted NHDP-MIB, also reports state, performance information and notifications. This additional state and performance information is useful to troubleshoot problems and performance issues during neighbor discovery.

Status of This Memo

Copyright Notice

Table of Contents

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 configuring parameters of the Neighborhood Discovery Protocol [NHDP] process on a router. The MIB defined in this memo, denoted NHDP-MIB, also reports state, performance information and notifications. This additional state and performance information is useful to troubleshoot problems and performance issues during neighbor discovery.

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 [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 [RFC2578], [RFC2579] and [RFC2580].

3. Conventions

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 [RFC2119].

4. Overview

[NHDP] allows a router in a Mobile Ad Hoc Network (MANET) to discover and track topological information of routers up to two hops away by virtue of exchanging HELLO messages. This information is useful for routers running various routing and multicast flooding protocols developed within the IETF MANET Working Group.

4.1. Terms

The following definitions apply throughout this document:

- Notification Objects - triggers and associated notification messages allowing for asynchronous tracking of pre-defined events on the managed router.

o Configuration Objects - switches, tables, objects which are initialized to default settings or set through the management interface defined by this MIB.

o State Objects - automatically generated values which define the current operating state of the NHDP protocol process in the router.

o Performance Objects - automatically generated values which help an administrator or automated tool to assess the performance of the NHDP protocol process on the router and the overall discovery performance within the MANET.

5. Structure of the MIB Module

This section presents the structure of the NHDP-MIB module. The MIB is arranged into the following structure:

o nhdpNotifications - objects defining NHDP-MIB notifications.

o nhdpObjects - defining objects within this MIB. The objects are arranged into the following groups:

  * Configuration Group - defining objects related to the configuration of the NHDP instance on the router.

  * State Group - defining objects which reflect the current state of the NHDP instance running on the router.

  * Performance Group - defining objects which are useful to a management station when characterizing the performance of NHDP on the router and in the MANET.

o nhdpConformance - defining the minimal and maximal conformance requirements for implementations of this MIB.

5.1. Notifications

This section describes the use of notifications, and mechanisms to enhance the ability to manage NHDP networks.

5.1.1. Introduction

Notifications can be emitted by an NHDP router as a reaction to a specific event. This allows a network manager to efficiently determine the source of problems or significant changes of configuration or topology, instead of polling a possibly large number of NHDP routers.

5.1.2. Notification Generation

When an exception event occurs, the application notifies the local agent, which sends a notification to the appropriate SNMP management stations. The message includes the notification type and may include a list of notification-specific variables. Section 7 contains, amongst others, the notification definitions, which includes the variable lists. At least one IP address of the NHDP router that originates the notification is included in the variable list so that the network manager may determine the source of the notification.

5.1.3. Limiting Frequency of Notifications

To limit the frequency of notifications, the following additional mechanisms are suggested, similar to those in [RFC4750]:

5.1.3.1. Ignoring Initial Activity

The majority of critical events occur when NHDP is enabled on a router, at which time the symmetric neighbors and two-hop neighbors of the NHDP router are discovered. During this initial period, a potential flood of notifications is unnecessary since the events are expected. To avoid unnecessary notifications, a router should not originate expected notifications until a certain time interval has elapsed, which is to be predefined by the network manager.

5.1.3.2. Throttling Traps

The mechanism for throttling the notifications is the same as in [RFC4750] (i.e. the amount of transmitted notifications per time is bounded).

Appropriate values for the window time and upper bound are to be selected by the network manager and depend on the deployment of the MANET.

5.1.3.3. One Notification per Event

Similar to the according mechanism in [RFC4750], only one notification is sent per event.

5.2. The Configuration Group

The NHDP router is configured with a set of controls. The authoritative list of configuration controls within the NHDP-MIB are found within the MIB module itself. Generally, an attempt was made in developing the NHDP-MIB module to support all configuration objects defined in [NHDP]. For all of the configuration parameters, the same constraints and default values of these parameters as defined in [NHDP] are followed.

5.3. The State Group

The State Group reports current state information of a router running [NHDP]. The NHDP-MIB State Group tables were designed to contain the complete set of state information defined within the information bases in [NHDP].

Two constructs, i.e., TEXTUAL CONVENTIONs, are defined in support of the tables in the State Group. These are NeighborIfIndex and NeighborRouterId. These are locally (to the NHDP router) defined, unique identifiers. They are used to define indexes to the appropriate State Group tables and to correlate table entries to interface addresses, interfaces and routers within the MANET. NeighborIfIndex is a unique identifier of discovered NHDP interfaces on all routers within the MANET. NeighborRouterId is a unique identifier of discovered NHDP routers within the MANET.

5.4. The Performance Group

The Performance Group reports values relevant to system performance. This section lists objects for NHDP performance monitoring, some of which are explicitly defined in the NHDP-MIB and others which are obtainable through a combination of base objects from this MIB and reports available through the REPORT-MIB [REPORT]. Throughout this section, those objects will be pointed out that are intended as base objects which are explicitly defined within this MIB and those objects which are derived through a combination of the base objects and capabilities offered by the REPORT-MIB.

Unstable neighbors or 2-hop neighbors and frequent changes of sets can have a negative influence on the performance of NHDP. The following objects allow management applications to acquire information related to the stability and performance of NHDP:

The following objects return statistics related to HELLO messages:

- o Total number of sent HELLO messages on an interface

  This is a Base Object.

  Object name: nhdpIfHelloMessageXmits

Object type: Counter32

- Total number of received HELLO messages on an interface

  This is a Base Object.

  Object name: nhdpIfHelloMessageRecvd

  Object type: Counter32

- Total number of sent periodic HELLO messages on an interface

  This is a Base Object.

  Object name: nhdpIfHelloMessagePeriodicXmits

  Object type: Counter32

- Total number of sent triggered HELLO messages on an interface

  This is a Base Object.

  Object name: nhdpIfHelloMessageTriggeredXmits

  Object type: Counter32

- Acquire history of HELLO message scheduling instances for a given time duration on an interface

  It is desirable to develop the history of the exact timestamps of each HELLO message that has been sent as well as the type of the message (triggered or periodical). The list of events starts at the given point of time t0 and ends at the given time t1.

  This is a Derived Object to be pulled from the REPORT-MIB. It is derived from, e.g., the nhdpIfHelloMessagePeriodicXmits Base Object from the NHDP-MIB along with the capabilities derived from the reportHistoryGroup from the REPORT-MIB.

- Histogram of the intervals between HELLO messages on an interface

  It is desirable to track the values (in a 2-dimensional array) that represent a histogram of intervals between HELLO messages, separated by periodic and triggered types. The histogram would display the distribution of intervals between two consecutive HELLOs of the same type (triggered or periodical) using a given bin size. It includes all HELLOs that have been sent after the given time t0 and before the given time t1.

This is a Derived Object to be pulled from the REPORT-MIB. It can be derived from, e.g., the nhdpIfHelloMessagePeriodicXmits Base Object from the NHDP-MIB along with the capabilities derived from the reportHistoryGroup from the REPORT-MIB. The network management application could convert this information into the desired histogram.

- Changes of the frequency of the message scheduling on an interface

  This object will divide the given time interval from t0 to t1 into a given number of equal parts. It then creates a histogram for each part and calculates the distances (e.g. using the Bhattacharyya distance) between each two adjacent histograms in time. A higher value between two histograms means more difference between the histograms. For instance, this is representative of an event that suddenly sends many triggered HELLO messages, whereas before there have been only very few such triggered messages.

  This is a Derived Object to be pulled from the REPORT-MIB, as previously discussed, albeit this is a bit more complex with respect to the management application.

- Average number of sent HELLO messages per second between the given time t0 and t1 on an interface

  This is a Derived Object to be pulled from the reportSampledGroup from the REPORT-MIB. It is derived from, e.g., the nhdpIfHelloMessageXmits Base Object.

- Average number of received HELLO messages per second on an interface between the given time t0 and t1

  This is a Derived Object to be pulled from the REPORT-MIB. See the previous discussion.

- Total accumulated size in octets of sent HELLO messages on an interface

  This is a Base Object.

  Object name: nhdpIfHelloMessageXmitAccumulatedSize

Object type: Counter32

- Total accumulated size in octets of received HELLO messages on an interface

  This is a Base Object.

  Object name: nhdpIfHelloMessageRecvdAccumulatedSize

  Object type: Counter32

- Average size in octets of sent HELLO messages between the given time t0 and t1 on an interface

  This is a Derived Object to be pulled from the reportSampledGroup from the REPORT-MIB. It is derived from, e.g., the nhdpIfHelloMessageRecvdAccumulatedSize Base Object from this NHDP-MIB.

- Average size in octets of received HELLO messages between the given time t0 and t1 on an interface

  This is a Derived Object to be pulled from the REPORT-MIB. See previous discussion.

- Total accumulated number of advertised symmetric neighbors in HELLOs on that interface.

  This is a Base Object.

  Object name: nhdpIfHelloMessageXmitAccumulatedSymmetricNeighborCount

  Object type: Counter32

- Total accumulated number of advertised heard neighbors in HELLOs on that interface

  This is a Base Object.

  Object name: nhdpIfHelloMessageXmitAccumulatedHeardNeighborCount

  Object type: Counter32

- Total accumulated number of advertised lost neighbors in HELLOs on that interface

This is a Base Object.

Object name: nhdpIfHelloMessageXmitAccumulatedLostNeighborCount

Object type: Counter32

o Number of expected packets from a given neighbor based on the packet sequence number on an interface

  This is a Base Object.

  Object name: nhdpDiscIfExpectedPackets

  Object type: Counter32

o Success rate of received packets (number of received packets divided by number of expected packets based on the packet sequence number)

  This is a Derived Object to be pulled from this NHDP-MIB. It is derived from, e.g., the nhdpDiscIfRecvdPackets and the nhdpDiscIfExpectedPackets Base Objects defined in this MIB. This metric is then computed by the network management application.

The following objects inspect the frequency of all Neighbor Set changes:

o Number of Neighbor Set changes

  This object counts each Neighbor Set change. A change occurs whenever a new Neighbor Tuple has been added, a Neighbor Tuple has been removed or any entry of a Neighbor Tuple has been modified.

  This is a Base Object.

  Object name: nhdpNibNeighborSetChanges

  Object type: Counter32

o Acquire history of Neighbor Set changes

  This object returns the history of the exact timestamps of each time the Neighbor Set has been changed.

This is a Derived Object to be pulled from the reportHistoryGroup of the REPORT-MIB. It is derived from the previously discussed Base Object.

- Histogram of the intervals between Neighbor Set changes

  Returns the values (in a 2-dimensional array) that represent a histogram of intervals between Neighbor Set changes.

  This is a Derived Object to be pulled from the reportHistoryGroup from the REPORT-MIB. It is derived from the previously discussed Base Object. The network management application would develop the histograms based upon lists obtained from the REPORT-MIB.

- Changes of the frequency of the Neighbor Set changes

  This object will divide the given time interval from t0 to t1 into a given number of equal parts. It then creates a histogram for each part and calculates the distances (e.g. using the Bhattacharyya distance) between each two adjacent histograms in time. A higher value between two histograms means more difference between the histograms.

  This is a Derived Object to be pulled from the reportHistoryGroup from the REPORT-MIB. It is derived from the previously discussed Base Object. The network management application could then compute the desired metrics.

The next objects examine the uptime of a given neighbor:

- Number of changes of a Neighbor Tuple

  Returns the number of changes to the given Neighbor Tuple.

  This is a Base Object.

  Object name: nhdpDiscNeighborNibNeighborSetChanges

  Object type: Counter32

- Neighbor uptime

  Returns the number of hundredths of a second since the Neighbor Tuple corresponding to the given neighbor exists.

This is a Base Object.

Object name: nhdpDiscNeighborNibNeighborSetUpTime

Object type: TimeTicks

- o Acquire history of change of onlink status of a given neighbor

  This object returns the history of the exact timestamps of each time the neighbor becomes onlink or offlink. A neighbor is said to become "onlink" if a new Neighbor Tuple is created that corresponds to the given neighbor. It becomes "offlink" if such a tuple has been deleted.

  This is a Derived Object to be pulled from the reportHistoryGroup of the REPORT-MIB. It is derived from, e.g., the nhdpDiscNeighborNibNeighborSetChanges Base Object defined in this MIB.

- o Histogram of the intervals between a change of the onlink status of a given neighbor

  Returns the values that represent a histogram of intervals between a change of the onlink status of a given neighbor. The histogram includes all changes that have been made after the given time t0 and before the given time t1.

  This is a Derived Object to be pulled from the reportHistoryGroup of the REPORT-MIB. It is derived from, e.g. the nhdpDiscNeighborNibNeighborSetChanges Base Object defined in this MIB. This object sits in the nhdpDiscNeighborSetPerfTable which is indexed by the nhdpDiscNeighborSetRouterId.

The following objects examine the stability of a neighbor. A neighbor is said to be unstable if it "flaps" frequently between several links. It is said to be stable if the set of Link Tuples that correspond to the given neighbor is stationary.

- o Count the changes of the interface over which a given neighbor can be reached

  This object counts each time the neighbor changes the interface over which it is reachable. That means that the corresponding Link Tuple of the given link moves from the Link Set of one interface to another interface.

This is a Base Object.

Object name: nhdpDiscNeighborNibNeighborSetReachableLinkChanges

Object type: Counter32

- Acquire history of changes of the interface over which a given neighbor can be reached

  This object returns the history of the exact timestamps of each time the neighbor changes the interface over which it is reachable. That means that the corresponding Link Tuple of the given link moves from the Link Set of one interface to another interface.

  This is a Derived Object to be pulled from the reportHistoryGroup of the REPORT-MIB. It is derived from, e.g., the nhdpDiscNeighborNibNeighborSetReachableLinkChanges Base Object. The network management could develop the desired histogram based upon the information retrieved from the REPORT-MIB.

- Histogram of the intervals between a change of the interface over which a given neighbor is reachable

  Returns the values that represent a histogram of intervals between a change of the interface over which a given neighbor is reachable after the given time t0 and before the given time t1.

  This is a Derived Object to be pulled from the reportHistoryGroup from the REPORT-MIB. It is derived from the previously discussed Base Object, nhdpDiscNeighborNibNeighborSetChanges counter. The network management application would develop the histograms based upon lists obtained from the REPORT-MIB.

The following objects inspect the stability of a given 2-hop neighbor:

- Count the changes of the N2_neighbor_iface_addr_list of a given 2-hop neighbor

  This object returns the count of the times the 2-hop neighbor changes its N2_neighbor_iface_addr_list, i.e. the neighbor over which it is reachable.

This is a Base Object.

Object name: nhdpIib2HopSetPerfChanges

Object type: Counter32

- o Acquire history of changes of the N2_neighbor_iface_addr_list of a given 2-hop neighbor

  This object returns the history of the exact timestamps of each time the 2-hop neighbor changes its N2_neighbor_iface_addr_list, i.e. the neighbor over which it is reachable.

  This is a Derived Object to be pulled from the reportHistoryGroup of the REPORT-MIB. It is derived from the previously discussed Base Object, nhdpIib2HopSetPerfChanges counter.

- o Histogram of the intervals between a change of a 2-hop neighbor's N2_neighbor_iface_addr_list

  Returns the values that represent a histogram of intervals between a change of the 2-hop neighbor's N2_neighbor_iface_addr_list after the given time t0 and before the given time t1.

  This is a Derived Object to be pulled from the reportHistoryGroup from the REPORT-MIB. It is derived from the previously discussed Base Object, nhdpIib2HopSetPerfChanges counter. The network management application would develop the histograms based upon lists obtained from the REPORT-MIB.

The next objects examine the uptime of a given 2-hop neighbor:

- o 2-hop Neighbor uptime

  Returns the number of hundredths of a second since the 2-Hop Tuple corresponding to the given 2-hop neighbor IP address was registered.

  This is a Base Object.

  Object name: nhdpIib2HopSetPerfUpTime

Object type: TimeTicks

- Acquire history of change of onlink status of a given 2-hop neighbor

  This object returns the history of the exact timestamps of each time the 2-hop neighbor becomes onlink or offlink. A 2-hop neighbor is said to become "onlink" if a new 2-hop Tuple is created that corresponds to the given 2-hop neighbor. It becomes "offlink" if such a tuple has been deleted.

  This is a Derived Object to be pulled from the reportHistoryGroup of the REPORT-MIB. It is derived from the previously discussed Base Object, nhdpIib2HopSetPerfChanges counter.

- Histogram of the intervals between a change of the onlink status of a given 2-hop neighbor

  Returns the values that represent a histogram of intervals between a change of the onlink status of a given 2-hop neighbor. The histogram includes all changes that have been made after the given time t0 and before the given time t1.

  This is a Derived Object to be pulled from the reportHistoryGroup from the REPORT-MIB. It is derived from the previously discussed Base Object, nhdpIib2HopSetPerfChanges counter. The network management application would develop the histograms based upon lists obtained from the REPORT-MIB.

6. Relationship to Other MIB Modules

This section specifies the relationship of the MIB modules contained in this document to other standards, particularly to standards containing other MIB modules. Definitions imported from other MIB modules and other MIB modules that SHOULD be implemented in conjunction with the MIB module contained within this document are identified in this section.

6.1. Relationship to the SNMPv2-MIB

The 'system' group in the SNMPv2-MIB [RFC3418] is defined as being mandatory for all systems, and the objects apply to the entity as a whole. The 'system' group provides identification of the management entity and certain other system-wide data. The NHDP-MIB does not duplicate those objects.

6.2. Relationship to Routing Protocol MIBs relying on the NHDP-MIB

[NHDP] allows routing protocols to rely on the neighborhood information that is discovered by means of HELLO message exchange. In order to allow for troubleshoot, fault isolate, and manage such routing protocols through a routing protocol MIB, it may be desired to align the State Group tables of the NHDP-MIB and the routing protocol MIB. This is accomplished through the definition of two TEXTUAL-CONVENTIONS in the NHDP-MIB: the NeighborInterfaceId and the NeighborRouterId. These object types are used to develop indexes into common NHDP-MIB and routing protocol State Group tables. These objects are locally significant but should be locally common to the NHDP-MIB and the routing protocol MIB implemented on a common networked router. This will allow for improved cross referencing of information across the two MIBs.

6.3. Relationship to the REPORT-MIB

This document describes several Performance Management metrics for the management of NHDP network routers. However, not all of these metrics are explicitly defined solely within the context of this NHDP-MIB. Some of these metrics are obtained through joint interaction between this MIB and the REPORT-MIB [REPORT]. This NHDP-MIB defines the minimum necessary objects (often of type COUNTER) which form the underlying basis for more sophisticated Performance Management reporting available in conjunction with the REPORT-MIB. See Section 5.4 for a discussion of the performance metrics for NHDP management.

6.4. MIB modules required for IMPORTS

The following NHDP-MIB module IMPORTS objects from SNMPv2-SMI [RFC2578], SNMPv2-TC [RFC2579], SNMPv2-CONF [RFC2580], IF-MIB [RFC2863], INET-ADDRESS-MIB [RFC4001], and SMIng [RFC3781].

7. Definitions

This section contains the MIB module defined by the specification.

```
NHDP-MIB DEFINITIONS ::= BEGIN

IMPORTS

    MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
    Counter32, Integer32, Unsigned32, mib-2, TimeTicks
                 FROM SNMPv2-SMI  --[RFC2578]

    TEXTUAL-CONVENTION, TruthValue, TimeStamp,
```

```
                 RowStatus
                 FROM SNMPv2-TC  --[RFC2579]

    MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
                 FROM SNMPv2-CONF  --[STD58]

    InetAddressType, InetAddress,
    InetAddressPrefixLength
                 FROM INET-ADDRESS-MIB  --[RFC4001]

    InterfaceIndexOrZero
                 FROM IF-MIB  --[RFC2863]

    Float32TC
                 FROM FLOAT-TC-MIB  --[RFCXXXX]
    ;

nhdpMIB MODULE-IDENTITY
       LAST-UPDATED "201101031000Z" -- January 3, 2011
       ORGANIZATION "IETF MANET working group"
       CONTACT-INFO
       "WG E-Mail: manet@ietf.org

        WG Chairs: ian.chakeres@gmail.com
                   jmacker@nrl.navy.mil


        Editors:   Ulrich Herberg
                   Ecole Polytechnique
                   LIX
                   91128 Palaiseau Cedex
                   France
                   +33 1 69 33 41 26
                   ulrich@herberg.name
                   http://www.herberg.name/

                   Robert G. Cole
                   US Army CERDEC
                   Space and Terrestrial Communications
                   328 Hopkins Road
                   Bldg 245, Room 16
                   Aberdeen Proving Ground, MD 21005
                   USA
                   +1 410 278-6779
                   robert.g.cole@us.army.mil
                   http://www.cs.jhu.edu/~rgcole/

                   Ian D Chakeres
```

```
                  CenGen
                  9250 Bendix Road North
                  Columbia, Maryland  21045
                  USA
                  ian.chakeres@gmail.com
                  http://www.ianchak.com/"

       DESCRIPTION
           "This NHDP-MIB module is applicable to routers
            implementing the Neighborhood Discovery Protocol
            defined in [RFC XXXX].

            Copyright (C) The IETF Trust (2009). This version
            of this MIB module is part of RFC xxxx; see the RFC
            itself for full legal notices."

       -- revision
       REVISION "201101031000Z" -- January 3, 2011
       DESCRIPTION
            "The tenth version of this MIB module,
             published as draft-ietf-manet-nhdp-mib-07.txt.
             Added FLOAT32TC from FLOAT-TC-MIB using this
             for representing the link quality parameters.
             Added a threshold (number) and window (time
             interval) within the nhdpNotificationsControl
             for the nhdpNbrStateChange, nhdp2HopNbrStateChange
             and nhdpIfRxBadPacket notifications.
            "
       REVISION "201011111000Z" -- November 11, 2010
       DESCRIPTION
            "The ninth version of this MIB module,
            published as draft-ietf-manet-nhdp-mib-06.txt.
            Corrected editorial issues, fixed some small
            bugs in the MIB."
       REVISION "201011081000Z" -- November 08, 2010
       DESCRIPTION
            "The eight version of this MIB module,
            published as draft-ietf-manet-nhdp-mib-05.txt.
            Cleaned up defaults and interdependence's
            between objects."
       REVISION    "201007071000Z"   -- July 07, 2010
       DESCRIPTION
         "The seventh version of this MIB module,
          published as draft-ietf-manet-nhdp-mib-04.txt.
          Cleaned up and condensed the textual material
          in the earlier sections of this draft.  Checked
          consistency with NHDP draft, i.e.,
          draft-ietf-manet-nhdp-12.txt."
```

```
       REVISION    "201003081000Z"   -- March 08, 2010
       DESCRIPTION
         "The sixth version of this MIB module,
          published as draft-ietf-manet-nhdp-mib-03.txt.
          Added the local nhdpIfIndex to the
          nhdpIibLinkSetTable."
       REVISION    "200911091000Z"   -- November 09, 2009
       DESCRIPTION
         "The fifth version of this MIB module,
          published as draft-ietf-manet-nhdp-mib-02.txt.
          Cleaned up a few things and updated to newest
          revision of NHDP draft."
       REVISION    "200910211000Z"   -- October 21, 2009
       DESCRIPTION
         "The fourth version of this MIB module,
          published as draft-ietf-manet-nhdp-mib-01.txt.
          Added objects pertaining to the performance
          group."
       REVISION    "200905031500Z"   -- May 3, 2009
       DESCRIPTION
         "The third version of this MIB module,
          published as draft-ietf-manet-nhdp-mib-00.txt.
          No major revisions to this draft.  Mainly rev'd
          as a new working group document.  But also cleaned
          syntax errors, typos and other issues discovered
          with 'smilint'."
       REVISION    "200902151500Z"   -- February 15, 2009
       DESCRIPTION
         "The second version of this MIB module,
          published as draft-cole-manet-nhdp-mib-01.txt.  Major
          update adding objects for configuration and state."
       REVISION    "200804251500Z"   -- April 25, 2008
       DESCRIPTION
         "The original version of this MIB module,
          published as draft-cole-manet-nhdp-mib-00.txt."
       -- RFC-Editor assigns XXXX
       ::= { mib-2 998 }   -- to be assigned by IANA

--
-- Top-Level Components of this MIB
--
nhdpNotifications OBJECT IDENTIFIER ::= { nhdpMIB 0 }
nhdpObjects       OBJECT IDENTIFIER ::= { nhdpMIB 1 }
nhdpConformance   OBJECT IDENTIFIER ::= { nhdpMIB 2 }


--
-- Textual Conventions
```

```
--

NeighborIfIndex ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS       current
    DESCRIPTION
        "A locally arbitrary unique identifier associated with an
        NHDP neighbor interface.

        All objects of type NeighborIfIndex are assigned by the agent
        out of a common number space. In other words, NeighborIfIndex
        values assigned to entries in one table must not overlap with
        NeighborIfIndex values assigned to entries in another
        table.

        The NeighborIfIndex defines a discovered interface of a 1-hop
        or 2-hop neighbor of the local router.  The agent identifies a
        unique neighbor interface through the receipt of an address
        lists advertised through an NHDP HELLO message.

        The value for each discovered neighbor interface must remain
        constant at least from one re-initialization of the entity's
        network management system to the next re-initialization, except
        that if an application is deleted and re-created.

        The specific value is meaningful only within a given SNMP
        entity. An NeighborIfIndex value must not be re-used until the
        next agent restart."
    SYNTAX       Unsigned32 (1..2147483647)


NeighborRouterId ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS       current
    DESCRIPTION
        "A locally arbitrary unique identifier associated with an
        NHDP discovered peer router.

        All objects of type NeighborRouterId are assigned by the agent
        out of a common number space.

        The NeighborRouterId defines a discovered NHDP peer of
        the local router. The agent identifies a
        unique neighbor interface through the receipt of an address
        list advertised through an NHDP HELLO message.

        The value for each discovered neighbor ID must remain
        constant at least from one re-initialization of the entity's
```

```
        network management system to the next re-initialization, except
        that if an application is deleted and re-created.

        The specific value is meaningful only within a given SNMP
        entity. An NeighborRouterId value must not be re-used until the
        next agent restart."
    SYNTAX       Unsigned32 (1..2147483647)



--
-- nhdpObjects
--

--    1) Configuration Objects Group
--    2) State Objects Group
--    3) Performance Objects Group



--
-- nhdpConfigurationObjGrp
--

-- Contains the NHDP objects which configure specific options
-- which determine the overall performance and operation of the
-- discovery protocol.


nhdpConfigurationObjGrp OBJECT IDENTIFIER ::= { nhdpObjects 1 }


   nhdpInterfaceTable  OBJECT-TYPE
      SYNTAX      SEQUENCE OF NhdpInterfaceEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "nhdpInterfaceTable describes the
          configuration of the interfaces of this NHDP router.
          The ifIndex is from the interfaces group
          defined in the Interfaces Group MIB.

          nhdpIfStatus provides the functionality
          expected by the NHDP in the Local Interface Base (LIB)
          Local Interface Set Table.  Hence, the Local Interface
          Set Table will not be defined below.

          The objects in this table are persistent and when
```

```
          written the entity SHOULD save the change to
          non-volatile storage."
      REFERENCE
         "RFC 2863 - The Interfaces Group MIB, McCloghrie,
          K., and F. Kastenholtz, June 2000."
   ::= { nhdpConfigurationObjGrp 1 }

   nhdpInterfaceEntry OBJECT-TYPE
      SYNTAX      NhdpInterfaceEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "nhdpInterfaceEntry describes one NHDP
          local interface configuration as indexed by
          its ifIndex as defined in the Standard MIB II
          Interface Table (RFC2863)."
      INDEX { nhdpIfIndex }
   ::= { nhdpInterfaceTable 1 }

   NhdpInterfaceEntry ::=
      SEQUENCE {
         nhdpIfIndex
            InterfaceIndexOrZero,
         nhdpIfStatus
            TruthValue,
         nhdpHelloInterval
            Unsigned32,
         nhdpHelloMinInterval
            Unsigned32,
         nhdpRefreshInterval
            Unsigned32,
         nhdpLHoldTime
            Unsigned32,
         nhdpHHoldTime
            Unsigned32,
         nhdpHystAcceptQuality
            Float32TC,
         nhdpHystRejectQuality
            Float32TC,
         nhdpInitialQuality
            Float32TC,
         nhdpInitialPending
            TruthValue,
         nhdpHpMaxJitter
            Unsigned32,
         nhdpHtMaxJitter
            Unsigned32,
         nhdpIfRowStatus
```

```
            RowStatus
         }

   nhdpIfIndex  OBJECT-TYPE
      SYNTAX      InterfaceIndexOrZero
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The ifIndex for this interface."
      ::= { nhdpInterfaceEntry 1 }

   nhdpIfStatus  OBJECT-TYPE
      SYNTAX      TruthValue
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "nhpdIfStatus indicates whether this interface is
         a MANET interface. A value of true(1) indicates
         that the interface is a MANET interface.  A value of
         false(2) indicates that the interface is not a MANET
         interface. This corresponds to the I_manet parameter
         in the Local Interface Set, which is omitted in this MIB
         due to the redundancy with the nhdpInterfaceTable."
      DEFVAL { 2 }
   ::= { nhdpInterfaceEntry 2 }


   --
   -- Interface Parameters - Message Intervals
   --

   nhdpHelloInterval  OBJECT-TYPE
      SYNTAX      Unsigned32
      UNITS       "milliseconds"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "nhpdHelloInterval corresponds to
         HELLO_INTERVAL of NHDP.

         The following constraint applies to this
         parameter:
             nhpdHelloInterval >= nhdpHelloMinInterval"
      REFERENCE
         "The NHDP draft.
          Section 5 on Protocol Parameters and
          Constraints."
      DEFVAL { 2000 }
```

```
   ::= { nhdpInterfaceEntry 3 }


   nhdpHelloMinInterval  OBJECT-TYPE
      SYNTAX      Unsigned32
      UNITS       "milliseconds"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "nhpdHelloMinInterval corresponds to
         HELLO_MIN_INTERVAL of NHDP."
      REFERENCE
         "The NHDP draft.
          Section 5 on Protocol Parameters and
          Constraints."
      DEFVAL { 500 }
   ::= { nhdpInterfaceEntry 4 }


   nhdpRefreshInterval  OBJECT-TYPE
      SYNTAX      Unsigned32
      UNITS       "milliseconds"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "nhpdRefreshInterval corresponds to
         REFRESH_INTERVAL of NHDP.

         The following constraint applies to this
         parameter:
             nhdpRefreshInterval >= nhdpHelloInterval"
      REFERENCE
         "The NHDP draft.
          Section 5 on Protocol Parameters and
          Constraints."
      DEFVAL { 2000 }
   ::= { nhdpInterfaceEntry 5 }

   --
   -- Interface Parameters - Information Validity times
   --

   nhdpLHoldTime  OBJECT-TYPE
      SYNTAX      Unsigned32
      UNITS       "milliseconds"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
```

```
         "nhdpLHoldTime corresponds to
         L_HOLD_TIME of NHDP."
      REFERENCE
         "The NHDP draft.
          Section 5 on Protocol Parameters and
          Constraints."
      DEFVAL { 6000 }
   ::= { nhdpInterfaceEntry 6 }

   nhdpHHoldTime  OBJECT-TYPE
      SYNTAX      Unsigned32
      UNITS       "milliseconds"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "nhdpHHoldTime corresponds to
         H_HOLD_TIME of NHDP.

         This object is persistent and when written
         the entity SHOULD save the change to
         non-volatile storage."
      REFERENCE
         "The NHDP draft.
          Section 5 on Protocol Parameters and
          Constraints."
      DEFVAL { 6000 }
   ::= { nhdpInterfaceEntry 7 }

   --
   -- Interface Parameters - Link Quality
   -- (is optional and settings define operation)
   --

   nhdpHystAcceptQuality  OBJECT-TYPE
      SYNTAX      Float32TC
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "nhdpHystAcceptQuality corresponds to
         HYST_ACCEPT of NHDP.

          The following constraint applies to this
          parameter:
              0 <= nhdpHystRejectQuality
                <= nhdpHystAcceptQuality <= 1.0"
      REFERENCE
         "The NHDP draft.
          Section 5 on Protocol Parameters and
```

```
          Constraints."
      -- DEFVAL { 1.0 }
   ::= { nhdpInterfaceEntry 8 }

   nhdpHystRejectQuality  OBJECT-TYPE
      SYNTAX      Float32TC
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "nhdpHystRejectQuality corresponds to
         HYST_REJECT of NHDP.

          The following constraint applies to this
          parameter:
              0 <= nhdpHystRejectQuality
                <= nhdpHystAcceptQuality <= 1.0"
      REFERENCE
         "The NHDP draft.
          Section 5 on Protocol Parameters and
          Constraints."
      -- DEFVAL { 0.0 }
   ::= { nhdpInterfaceEntry 9 }

   nhdpInitialQuality  OBJECT-TYPE
      SYNTAX      Float32TC
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "nhdpInitialQuality corresponds to
         INITIAL_QUALITY of NHDP.

          The following constraint applies to this
          parameter:
              0 <= nhdpInitialQuality <= 1.0"
      REFERENCE
         "The NHDP draft.
          Section 5 on Protocol Parameters and
          Constraints."
      -- DEFVAL { 1.0 }
   ::= { nhdpInterfaceEntry 10 }



   nhdpInitialPending  OBJECT-TYPE
      SYNTAX      TruthValue
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
```

```
         "nhdpInitialPending corresponds to
         INITIAL_PENDING of NHDP."
      REFERENCE
         "The NHDP draft.
          Section 5 on Protocol Parameters and
          Constraints."
      DEFVAL { 2 }   -- i.e. false
   ::= { nhdpInterfaceEntry 11 }

   --
   -- Interface Parameters - Jitter
   --
   nhdpHpMaxJitter  OBJECT-TYPE
      SYNTAX      Unsigned32
      UNITS       "milliseconds"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "nhdpHpMaxJitter corresponds to
         HP_MAXJITTER of NHDP."
      REFERENCE
         "The NHDP draft.
          Section 5 on Protocol Parameters and
          Constraints."
      DEFVAL { 500 }
   ::= { nhdpInterfaceEntry 12 }


   nhdpHtMaxJitter  OBJECT-TYPE
      SYNTAX      Unsigned32
      UNITS       "milliseconds"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "nhdpHtMaxJitter corresponds to
         HT_MAXJITTER of NHDP."
      REFERENCE
         "The NHDP draft.
          Section 5 on Protocol Parameters and
          Constraints."
      DEFVAL { 500 }
   ::= { nhdpInterfaceEntry 13 }

   nhdpIfRowStatus  OBJECT-TYPE
      SYNTAX      RowStatus
      MAX-ACCESS  read-create
      STATUS      current
      DESCRIPTION
```

```
        "This object permits management of the table
        by facilitating actions such as row creation,
        construction, and destruction. The value of
        this object has no effect on whether other
        objects in this conceptual row can be
        modified."
     REFERENCE
        "The NHDP draft."
   ::= { nhdpInterfaceEntry 14 }


   --
   -- Router Parameters - Information Validity Time
   --
   nhdpNHoldTime  OBJECT-TYPE
     SYNTAX      Unsigned32
     UNITS       "milliseconds"
     MAX-ACCESS  read-write
     STATUS      current
     DESCRIPTION
        "nhdpNHoldTime corresponds to
        N_HOLD_TIME of NHDP.

         This object is persistent and when written
         the entity SHOULD save the change to
         non-volatile storage."
     REFERENCE
        "The NHDP draft.
         Section 5 on Protocol Parameters and
         Constraints."
     DEFVAL { 6000 }
   ::= { nhdpConfigurationObjGrp 2 }


   nhdpIHoldTime  OBJECT-TYPE
     SYNTAX      Unsigned32
     UNITS       "milliseconds"
     MAX-ACCESS  read-write
     STATUS      current
     DESCRIPTION
        "nhdpIHoldTime corresponds to
        I_HOLD_TIME of NHDP.

         This object is persistent and when written
         the entity SHOULD save the change to
         non-volatile storage."
     REFERENCE
        "The NHDP draft.
```

```
          Section 5 on Protocol Parameters and
          Constraints."
      DEFVAL { 6000 }
   ::= { nhdpConfigurationObjGrp 3 }




--
-- nhdpStateObjGrp
--

-- Contains information describing the current state of the NHDP
-- process.

nhdpStateObjGrp    OBJECT IDENTIFIER ::= { nhdpObjects 2 }

   -- Two new constructs have been defined in this MIB for
   -- indexing into the following
   -- tables and indexing into other tables in other MIBs.
   -- The NeighborIfIndex defines a unique (to the local router)
   -- index referencing a discovered interface on another
   -- router within the MANET. The NeighborRouterId defines a
   -- unique (to the local router) index referencing a discovered
   -- router within the MANET.



   -- This table is indexed by an IpAddr associated with
   -- NeighborIfIndex.  Multiple addresses can be associated
   -- with a given NeighborIfIndex.  Each NeighborIfIndex is
   -- associated with a NeighborRouterId.  Throughout this MIB,
   -- the NeighborIfIndex and the NeighborRouterId are used
   -- to define the set of IpAddrs related to the interface
   -- in discussion.

   nhdpUpTime OBJECT-TYPE
       SYNTAX TimeTicks
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
             "The number of hundredths of a second since the
             current NHDP process was initialized."
        REFERENCE
          "The NHDP draft."
    ::= { nhdpStateObjGrp 1 }
```

```
   nhdpDiscIfSetTable OBJECT-TYPE
      SYNTAX       SEQUENCE OF NhdpDiscIfSetEntry
       MAX-ACCESS   not-accessible
       STATUS       current
       DESCRIPTION
           "A router's set of discovered interfaces on
            neighboring routers."
       REFERENCE
          "The NHDP draft."
    ::= { nhdpStateObjGrp 2 }


    nhdpDiscIfSetEntry  OBJECT-TYPE
       SYNTAX      NhdpDiscIfSetEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
          "The entries include the nhdpDiscRouterId of
           the discovered router, the nhdpDiscIfIndex
           of the discovered interface and the
           current set of addresses associated
           with this neighbor interface.  The
           nhdpDiscIfIndex has to uniquely identify
           the remote interface address sets.  It
           does not need to be unique across the MANET.
           It must be unique within this router."
       REFERENCE
          "This document."
       INDEX { nhdpDiscIfSetIpAddrType,
               nhdpDiscIfSetIpAddr }
    ::= { nhdpDiscIfSetTable 1 }

    NhdpDiscIfSetEntry ::=
       SEQUENCE {
          nhdpDiscIfSetRouterId
            NeighborRouterId,
          nhdpDiscIfSetIndex
            NeighborIfIndex,
          nhdpDiscIfSetIpAddrType
            InetAddressType,
          nhdpDiscIfSetIpAddr
            InetAddress,
          nhdpDiscIfSetIpAddrPrefixLen
            InetAddressPrefixLength
         }

   nhdpDiscIfSetRouterId  OBJECT-TYPE
      SYNTAX      NeighborRouterId
```

```
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The NHDP router ID (locally created)
          of a neighboring router.  Used for cross
          indexing into other NHDP tables and other
          MIBs."
      REFERENCE
         "This document."
   ::= { nhdpDiscIfSetEntry 1 }

   nhdpDiscIfSetIndex  OBJECT-TYPE
      SYNTAX      NeighborIfIndex
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The NHDP interface index (locally created)
          of a neighbor's interface.  Used for cross
          indexing into other NHDP tables and other
          MIBs."
      REFERENCE
         "This document."
   ::= { nhdpDiscIfSetEntry 2 }

   nhdpDiscIfSetIpAddrType  OBJECT-TYPE
      SYNTAX      InetAddressType
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "The type of the nhdpDiscIfSetIpAddr
          in the InetAddress MIB [RFC 4001]."
      REFERENCE
         "The NHDP draft."
   ::= { nhdpDiscIfSetEntry 3 }

   nhdpDiscIfSetIpAddr  OBJECT-TYPE
      SYNTAX      InetAddress
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "The nhdpDiscIfSetIpAddr is a
          recently used address of a neighbor
          of this router."
      REFERENCE
         "The NHDP draft."
   ::= { nhdpDiscIfSetEntry 4 }

   nhdpDiscIfSetIpAddrPrefixLen  OBJECT-TYPE
```

```
      SYNTAX      InetAddressPrefixLength
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "Indicates the number of leading one bits that form the
          mask to be logical-ANDed with the destination address
          before being compared to the value in the
          nhdpDiscIfSetAddr field.  If the resulting
          address block is contained in a block in this
          table, then a match should be returned."
      REFERENCE
         "The NHDP draft."
   ::= { nhdpDiscIfSetEntry 5 }



   -- An NHDP router's Local Information Base (LIB)

      -- Note: Local IF Set Table is not specified in this
      --       MIB because the table would be redundant with
      --       information in nhdpInterfaceTable.



      -- Removed Interface Addr Set Table
      -- Entry (foreach Addr): (IfAddrRemoved,
      --                        ExpirationTime)

    nhdpLibRemovedIfAddrSetTable OBJECT-TYPE
       SYNTAX       SEQUENCE OF NhdpLibRemovedIfAddrSetEntry
       MAX-ACCESS   not-accessible
       STATUS       current
       DESCRIPTION
          "A router's Removed Interface Address Set records
          network addresses which were recently used as local
          interface network addresses.  If a router's interface
          network addresses are immutable then the Removed
          Interface Address Set is always empty and MAY be omitted.
          It consists of Removed Interface Address Tuples, one
          per network address."
       REFERENCE
          "The NHDP draft."
    ::= { nhdpStateObjGrp 3 }

    nhdpLibRemovedIfAddrSetEntry  OBJECT-TYPE
       SYNTAX      NhdpLibRemovedIfAddrSetEntry
       MAX-ACCESS  not-accessible
```

```
       STATUS      current
       DESCRIPTION
          "A router's Removed Interface Address Set consists
          of Removed Interface Address Tuples, one per network
          address:

          (IR_local_iface_addr, IR_time)

          The association between these addrs and
          the router's Interface is found in the
          Standard MIB II's IP address table
          (RFC1213)."
       REFERENCE
          "The NHDP draft."
       INDEX { nhdpLibRemovedIfAddrSetIpAddrType,
               nhdpLibRemovedIfAddrSetIpAddr }
    ::= { nhdpLibRemovedIfAddrSetTable 1 }

    NhdpLibRemovedIfAddrSetEntry ::=
       SEQUENCE {
          nhdpLibRemovedIfAddrSetIpAddrType
            InetAddressType,
          nhdpLibRemovedIfAddrSetIpAddr
            InetAddress,
          nhdpLibRemovedIfAddrSetIpAddrPrefixLen
            InetAddressPrefixLength,
          nhdpLibRemovedIfAddrSetIfIndex
            InterfaceIndexOrZero,
          nhdpLibRemovedIfAddrSetIrTime
            TimeStamp
         }

   nhdpLibRemovedIfAddrSetIpAddrType  OBJECT-TYPE
      SYNTAX      InetAddressType
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "The type of the nhdpLibRemovedIfAddrSetIpAddr
          in the InetAddress MIB [RFC 4001]."
      REFERENCE
         "The NHDP draft."
   ::= { nhdpLibRemovedIfAddrSetEntry 1 }

   nhdpLibRemovedIfAddrSetIpAddr  OBJECT-TYPE
      SYNTAX      InetAddress
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
```

```
         "nhdpLibRemovedIfAddrSetAddr is a
          recently used address of an interface of
          this router."
      REFERENCE
         "The NHDP draft."
   ::= { nhdpLibRemovedIfAddrSetEntry 2 }

   nhdpLibRemovedIfAddrSetIpAddrPrefixLen  OBJECT-TYPE
      SYNTAX      InetAddressPrefixLength
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "Indicates the number of leading one bits that form the
          mask to be logical-ANDed with the address
          to determine the network address to which
          this interface is attached."
      REFERENCE
         "The NHDP draft."
   ::= { nhdpLibRemovedIfAddrSetEntry 3 }


     nhdpLibRemovedIfAddrSetIfIndex  OBJECT-TYPE
        SYNTAX      InterfaceIndexOrZero
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
           "Specifies the local IfIndex from which this
            IP address was recently removed."
        REFERENCE
           "The NHDP draft."
     ::= { nhdpLibRemovedIfAddrSetEntry 4 }

     nhdpLibRemovedIfAddrSetIrTime  OBJECT-TYPE
        SYNTAX      TimeStamp
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
           "Specifies when this Tuple expires and MUST be removed
            from this table."
        REFERENCE
           "The NHDP draft."
     ::= { nhdpLibRemovedIfAddrSetEntry 5 }




   -- Interface Information Base (IIB)
```

```
   --
   -- NHDP Interface Information Base (IIB)
   --

   --     IIB Link Set
   --         Entry (foreach 1-H neighbor): (NeighborIfAddrList,
   --                                        HeardTime,
   --                                        SymTime,
   --                                        Quality,
   --                                        Pending,
   --                                        Lost,
   --                                        ExpireTime)

   nhdpIibLinkSetTable OBJECT-TYPE
      SYNTAX       SEQUENCE OF NhdpIibLinkSetEntry
      MAX-ACCESS   not-accessible
      STATUS       current
      DESCRIPTION
          "A Link Set of an interface records links from
           other routers which are, or recently
           were, 1-hop neighbors."
      REFERENCE
         "The NHDP draft."
   ::= { nhdpStateObjGrp 4 }

    nhdpIibLinkSetEntry  OBJECT-TYPE
       SYNTAX      NhdpIibLinkSetEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
          "A Link Set consists of Link Tuples, each
           representing a single link indexed by the
           local and remote interface pair:

           (L_neighbor_iface_addr_list, L_HEARD_time,
            L_SYM_time, L_quality, L_pending,
            L_lost, L_time).

            Note that L_quality is not included in the
            entries below, because updates may be
            required too frequently."
       REFERENCE
          "This document."
       INDEX { nhdpIfIndex,
               nhdpIibLinkSet1HopIfIndex }
    ::= { nhdpIibLinkSetTable 1 }

    NhdpIibLinkSetEntry ::=
```

```
        SEQUENCE {
           nhdpIibLinkSet1HopIfIndex
             NeighborIfIndex,
           nhdpIibLinkSetIfIndex
             InterfaceIndexOrZero,
           nhdpIibLinkSetLHeardTime
             TimeStamp,
           nhdpIibLinkSetLSymTime
             TimeStamp,
           nhdpIibLinkSetLPending
             TruthValue,
           nhdpIibLinkSetLLost
             TruthValue,
           nhdpIibLinkSetLTime
             TimeStamp
          }

    nhdpIibLinkSet1HopIfIndex  OBJECT-TYPE
       SYNTAX      NeighborIfIndex
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
          "nhdpIibLinkSet1HopIfIndex is
           the value of the NeighborIfIndex (from
           nhdpDiscIfSetTable). This
           object is repeated here to support
           table walks to view the set of neighbors
           of this router."
       REFERENCE
          "The NHDP draft."
    ::= { nhdpIibLinkSetEntry 1 }

    nhdpIibLinkSetIfIndex  OBJECT-TYPE
       SYNTAX      InterfaceIndexOrZero
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "nhdpIibLinkSetIfIndex is
           the local router's interface
           index associated with the symmetric
           link to this entrie's neighbor
           interface.

           The set of IP addresses associated with
           this neighbor's interface is found in
           nhdpDiscIfSetTable."
       REFERENCE
          "The NHDP draft."
```

```
   ::= { nhdpIibLinkSetEntry 2 }

   nhdpIibLinkSetLHeardTime  OBJECT-TYPE
      SYNTAX      TimeStamp
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "nhdpIibLinkSetLHeardTime corresponds
         to L_HEARD_time of NHDP."
      REFERENCE
         "The NHDP draft."
   ::= { nhdpIibLinkSetEntry 3 }

   nhdpIibLinkSetLSymTime  OBJECT-TYPE
      SYNTAX      TimeStamp
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "nhdpIibLinkSetLSymTime corresponds
         to L_SYM_time of NHDP."
      REFERENCE
         "The NHDP draft."
   ::= { nhdpIibLinkSetEntry 4 }

   nhdpIibLinkSetLPending  OBJECT-TYPE
      SYNTAX      TruthValue
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "nhdpIibLinkSetLPending corresponds
         to L_pending of NHDP"
      REFERENCE
         "The NHDP draft."
   ::= { nhdpIibLinkSetEntry 5 }

   nhdpIibLinkSetLLost  OBJECT-TYPE
      SYNTAX      TruthValue
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "nhdpIibLinkSetLLost corresponds
         to L_lost of NHDP"
      REFERENCE
         "The NHDP draft."
   ::= { nhdpIibLinkSetEntry 6 }

   nhdpIibLinkSetLTime  OBJECT-TYPE
      SYNTAX      TimeStamp
```

```
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "nhdpIibLinkSetLTime specifies
          when this Tuple expires and MUST
          be removed."
      REFERENCE
         "The NHDP draft."
   ::= { nhdpIibLinkSetEntry 7 }




  --
  --     IIB 2-Hop Set
  --         Entry (foreach IF on a 2-H neighbor):
  --                                (1NeighIfAddrList,
  --                                 2NeighIfAddr,
  --                                 ExpireTime)
  --
    nhdpIib2HopSetTable OBJECT-TYPE
       SYNTAX       SEQUENCE OF NhdpIib2HopSetEntry
       MAX-ACCESS   not-accessible
       STATUS       current
       DESCRIPTION
           "A 2-Hop Set of an interface records network
           addresses of symmetric 2-hop neighbors, and
           the symmetric links to symmetric 1-hop neighbors
           through which these symmetric 2-hop neighbors
           can be reached.  It consists of 2-Hop Tuples,
           each representing a single network address of
           a symmetric 2-hop neighbor, and a single MANET
           interface of a symmetric 1-hop neighbor.

           (N2_neighbor_iface_addr_list,
            N2_2hop_addr, N2_time)."
       REFERENCE
          "The NHDP draft."
    ::= { nhdpStateObjGrp 5 }

    nhdpIib2HopSetEntry  OBJECT-TYPE
       SYNTAX      NhdpIib2HopSetEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
          "The entries include the 2-hop neighbor addresses,
           which act as the table index, and associated
```

```
           1-hop symmetric link address set, designated
           through nhdpDiscIfIndex, and an expiration time."
       REFERENCE
          "This document."
       INDEX { nhdpIib2HopSetIpAddressType,
               nhdpIib2HopSetIpAddress }
    ::= { nhdpIib2HopSetTable 1 }

    NhdpIib2HopSetEntry ::=
       SEQUENCE {
          nhdpIib2HopSetIpAddressType
            InetAddressType,
          nhdpIib2HopSetIpAddress
            InetAddress,
          nhdpIib2HopSet1HopIfIndex
            NeighborIfIndex,
          nhdpIib2HopSetN2Time
            TimeStamp
         }


   nhdpIib2HopSetIpAddressType  OBJECT-TYPE
      SYNTAX      InetAddressType
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The type of the nhdpIib2HopSetIpAddress
          in the InetAddress MIB [RFC 4001]."
      REFERENCE
         "The NHDP draft."
   ::= { nhdpIib2HopSetEntry 1 }

   nhdpIib2HopSetIpAddress  OBJECT-TYPE
      SYNTAX      InetAddress
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "nhdpIib2HopSetIpAddr corresponds
         to N2_2hop_addr of NHDP."
      REFERENCE
         "The NHDP draft."
   ::= { nhdpIib2HopSetEntry 2 }

   nhdpIib2HopSet1HopIfIndex  OBJECT-TYPE
      SYNTAX      NeighborIfIndex
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
```

```
         "nhdpIib2HopSet1HopIfIndex is
          NeighborIfIndex of the one hop
          neighbor which communicated the ipAddress
          of the 2-hop neighbor in this row entry."
      REFERENCE
         "The NHDP draft."
   ::= { nhdpIib2HopSetEntry 3 }

   nhdpIib2HopSetN2Time  OBJECT-TYPE
      SYNTAX      TimeStamp
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "nhdpIib2HopSetN2Time specifies
          when this column entry expires and
          MUST be removed."
      REFERENCE
         "The NHDP draft."
   ::= { nhdpIib2HopSetEntry 4 }




   --
   -- Neighbor Information Base (NIB)
   --
   -- Each router maintains a Neighbor Information Base
   -- that records information about addresses of
   -- current and recently symmetric 1-hop neighbors.


   --     NIB Neighbor Set
   --         Entry (foreach 1-H Neighbor):
   --              (AllIfAddrListOfIhNeighbor,
   --               SymmetricIndicator)
   --     The NIB Neighbor Set Table is small because
   --     most of the corresponding information is found
   --     in the nhdpDiscoveredIfTable above.
   --
   nhdpNibNeighborSetTable OBJECT-TYPE
      SYNTAX       SEQUENCE OF NhdpNibNeighborSetEntry
      MAX-ACCESS   not-accessible
      STATUS       current
      DESCRIPTION
          "A router's Neighbor Set records all network
          addresses of each 1-hop neighbor."
      REFERENCE
         "The NHDP draft."
```

```
   ::= { nhdpStateObjGrp 6 }

   nhdpNibNeighborSetEntry  OBJECT-TYPE
      SYNTAX      NhdpNibNeighborSetEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "A router's Neighbor Set consists
           of Neighbor Tuples, each representing
           a single 1-hop neighbor:

           (N_neighbor_addr_list,
            N_symmetric)"
      REFERENCE
         "This document."
      INDEX { nhdpDiscIfSetRouterId }
   ::= { nhdpNibNeighborSetTable 1 }

    NhdpNibNeighborSetEntry ::=
       SEQUENCE {
          nhdpNibNeighborSetNSymmetric
            TruthValue
         }

   nhdpNibNeighborSetNSymmetric  OBJECT-TYPE
      SYNTAX      TruthValue
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "nhdpNibNeighborNSymmetric corresponds
         to N_symmetric of NHDP."
      REFERENCE
         "The NHDP draft."
   ::= { nhdpNibNeighborSetEntry 1 }



  --     Lost Neighbor Set
  --         Entry ( foreach IF foreach 1-H Neighbor): (IfAddr,
  --                                                   ExpireTime)
  --
    nhdpNibLostNeighborSetTable OBJECT-TYPE
       SYNTAX       SEQUENCE OF NhdpNibLostNeighborSetEntry
       MAX-ACCESS   not-accessible
       STATUS       current
       DESCRIPTION
           "A router's Lost Neighbor Set records network
           addresses of routers which recently were
```

```
            symmetric 1-hop neighbors, but which are now
            advertised as lost."
        REFERENCE
           "The NHDP draft."
     ::= { nhdpStateObjGrp 7 }

     nhdpNibLostNeighborSetEntry  OBJECT-TYPE
        SYNTAX      NhdpNibLostNeighborSetEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
           "A router's Lost Neighbor Set consists of
           Lost Neighbor Tuples, each representing a
           single such network address:

           (NL_neighbor_addr, NL_time)"
        REFERENCE
           "This document."
        INDEX { nhdpDiscIfSetRouterId }
     ::= { nhdpNibLostNeighborSetTable 1 }

     NhdpNibLostNeighborSetEntry ::=
        SEQUENCE {
           nhdpNibLostNeighborSetNLTime
             TimeStamp
          }

    nhdpNibLostNeighborSetNLTime  OBJECT-TYPE
       SYNTAX      TimeStamp
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "nhdpNibLostNeighborSetNLTime
           specifies when this Tuple expires
           and MUST be removed."
       REFERENCE
          "The NHDP draft."
    ::= { nhdpNibLostNeighborSetEntry 1 }



 --
 -- nhdpPerformanceObjGrp
 --

 -- Contains objects which help to characterize the performance of
 -- the NHDP process, typically counters.
 --
```

```
nhdpPerformanceObjGrp OBJECT IDENTIFIER ::= { nhdpObjects 3 }

  --
  -- Objects per local interface
  --

  nhdpInterfacePerfTable  OBJECT-TYPE
      SYNTAX      SEQUENCE OF NhdpInterfacePerfEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "This table summarizes performance objects that are
          measured per local NHDP interface."
      REFERENCE
         "The NHDP draft."
   ::= { nhdpPerformanceObjGrp 1 }

   nhdpInterfacePerfEntry OBJECT-TYPE
      SYNTAX      NhdpInterfacePerfEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "A single entry contains performance counters for
          a local NHDP interface."
      INDEX { nhdpIfIndex }
   ::= { nhdpInterfacePerfTable 1 }

   NhdpInterfacePerfEntry ::=
      SEQUENCE {
         nhdpIfHelloMessageXmits
            Counter32,
         nhdpIfHelloMessageRecvd
            Counter32,
         nhdpIfHelloMessageXmitAccumulatedSize
            Counter32,
         nhdpIfHelloMessageRecvdAccumulatedSize
            Counter32,
         nhdpIfHelloMessageTriggeredXmits
            Counter32,
         nhdpIfHelloMessagePeriodicXmits
            Counter32,
         nhdpIfHelloMessageXmitAccumulatedSymmetricNeighborCount
            Counter32,
         nhdpIfHelloMessageXmitAccumulatedHeardNeighborCount
            Counter32,
         nhdpIfHelloMessageXmitAccumulatedLostNeighborCount
            Counter32
         }
```

```
   nhdpIfHelloMessageXmits  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter is incremented each time a HELLO
         message has been transmitted on that interface."
   ::= { nhdpInterfacePerfEntry 1 }

   nhdpIfHelloMessageRecvd  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter is incremented each time a
         HELLO message has been received on that interface."
   ::= { nhdpInterfacePerfEntry 2 }

   nhdpIfHelloMessageXmitAccumulatedSize  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter is incremented by the number of octets in
         a HELLO message each time a
         HELLO message has been sent."
   ::= { nhdpInterfacePerfEntry 3 }

   nhdpIfHelloMessageRecvdAccumulatedSize  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter is incremented by the number of octets in
         a HELLO message each time a
         HELLO message has been received."
   ::= { nhdpInterfacePerfEntry 4 }

   nhdpIfHelloMessageTriggeredXmits  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter is incremented each time a triggered
         HELLO message has been sent."
   ::= { nhdpInterfacePerfEntry 5 }

   nhdpIfHelloMessagePeriodicXmits  OBJECT-TYPE
```

```
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter is incremented each time a periodic
         HELLO message has been sent."
   ::= { nhdpInterfacePerfEntry 6 }

   nhdpIfHelloMessageXmitAccumulatedSymmetricNeighborCount  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter is incremented by the number of advertised
         symmetric neighbors in a HELLO each time a HELLO
         message has been sent."
   ::= { nhdpInterfacePerfEntry 7 }

   nhdpIfHelloMessageXmitAccumulatedHeardNeighborCount  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter is incremented by the number of advertised
         heard neighbors in a HELLO each time a HELLO
         message has been sent."
   ::= { nhdpInterfacePerfEntry 8 }

   nhdpIfHelloMessageXmitAccumulatedLostNeighborCount  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter is incremented by the number of advertised
         lost neighbors in a HELLO each time a HELLO
         message has been sent."
   ::= { nhdpInterfacePerfEntry 9 }



  --
  -- Objects per discovered neighbor interface
  --
  nhdpDiscIfSetPerfTable OBJECT-TYPE
       SYNTAX       SEQUENCE OF NhdpDiscIfSetPerfEntry
       MAX-ACCESS   not-accessible
       STATUS       current
       DESCRIPTION
```

```
          "A router's set of performance properties for
          each discovered interface of a neighbor."
       REFERENCE
          "The NHDP draft."
   ::= { nhdpPerformanceObjGrp 2 }


   nhdpDiscIfSetPerfEntry  OBJECT-TYPE
       SYNTAX      NhdpDiscIfSetPerfEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
          "There is an entry for each discovered
          interface of a neighbor."
       REFERENCE
          "This document."
       INDEX { nhdpDiscIfSetIndex }
   ::= { nhdpDiscIfSetPerfTable 1 }

    NhdpDiscIfSetPerfEntry ::=
       SEQUENCE {
          nhdpDiscIfRecvdPackets
            Counter32,
          nhdpDiscIfExpectedPackets
            Counter32
         }

   nhdpDiscIfRecvdPackets  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "This counter increments each
         time this router receives a packet from that interface
         of the neighbor."
      REFERENCE
         "The NHDP draft."
   ::= { nhdpDiscIfSetPerfEntry 1 }

   nhdpDiscIfExpectedPackets  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "This counter increments by the number
          of missed packets from this neighbor based
          on the packet sequence number each time this
          router receives a packet from that interface
```

```
          of the neighbor."
      REFERENCE
         "The NHDP draft."
   ::= { nhdpDiscIfSetPerfEntry 2 }



   --
   -- Objects concerning the neighbor set
   --
   nhdpNibNeighborSetChanges  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "This counter increments each time the Neighbor Set changes.
         A change occurs whenever a new Neighbor Tuple has been
         added, a Neighbor Tuple has been removed or any entry of
         a Neighbor Tuple has been modified."
   ::= { nhdpPerformanceObjGrp 3 }



   --
   -- Objects per discovered neighbor
   --
   nhdpDiscNeighborSetPerfTable OBJECT-TYPE
      SYNTAX       SEQUENCE OF NhdpDiscNeighborSetPerfEntry
       MAX-ACCESS   not-accessible
       STATUS       current
       DESCRIPTION
          "A router's set of discovered neighbors and
           their properties."
       REFERENCE
          "The NHDP draft."
   ::= { nhdpPerformanceObjGrp 4 }

   nhdpDiscNeighborSetPerfEntry  OBJECT-TYPE
       SYNTAX      NhdpDiscNeighborSetPerfEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
          "The entries include the nhdpDiscRouterId of
           the discovered router, as well as performance
           objects related to changes of the Neighbor
           Set."
       REFERENCE
          "This document."
```

```
       INDEX { nhdpDiscIfSetRouterId }
   ::= { nhdpDiscNeighborSetPerfTable 1 }

    NhdpDiscNeighborSetPerfEntry ::=
       SEQUENCE {
          nhdpDiscNeighborNibNeighborSetChanges
            Counter32,
          nhdpDiscNeighborNibNeighborSetUpTime
            TimeTicks,
          nhdpDiscNeighborNibNeighborSetReachableLinkChanges
            Counter32
         }

   nhdpDiscNeighborNibNeighborSetChanges  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "This counter increments each time the neighbor becomes
          onlink or offlink.  A neighbor is said to become
          'onlink' if a new nhdpNibNeighborSetEntry is created
          for a particular nhdpNibNeighborSetRouterId. It becomes
          'offlink' if the entry for that neighbor has been deleted."
      REFERENCE
         "The NHDP draft."
   ::= { nhdpDiscNeighborSetPerfEntry 1 }


   nhdpDiscNeighborNibNeighborSetUpTime  OBJECT-TYPE
      SYNTAX      TimeTicks
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "This object returns the time in hundredths of a second since
         the neighbor becomes 'onlink'.  A neighbor is
         said to become 'onlink' if a new nhdpNibNeighborSetEntry
         is created for a particular nhdpNibNeighborSetRouterId.
         It becomes 'offlink' if the entry for that neighbor
         has been deleted."
      REFERENCE
         "This document."
   ::= { nhdpDiscNeighborSetPerfEntry 2 }

   nhdpDiscNeighborNibNeighborSetReachableLinkChanges  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
```

```
         "This counter increments each
         time the neighbor changes the interface over which it is
         reachable.  That means that the corresponding Link Tuple of the
         given link moves from the Link Set of one interface to another
         interface."
      REFERENCE
         "The NHDP draft."
   ::= { nhdpDiscNeighborSetPerfEntry 3 }




   --
   -- Objects per discovered 2-hop neighbor
   --
      nhdpIib2HopSetPerfTable OBJECT-TYPE
       SYNTAX       SEQUENCE OF NhdpIib2HopSetPerfEntry
       MAX-ACCESS   not-accessible
       STATUS       current
       DESCRIPTION
           "This table contains performance objects per
           discovered 2-hop neighbor."
       REFERENCE
          "The NHDP draft."
    ::= { nhdpPerformanceObjGrp 5 }

    nhdpIib2HopSetPerfEntry  OBJECT-TYPE
       SYNTAX      NhdpIib2HopSetPerfEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
          "The entries contain performance objects per
           discovered 2-hop neighbor."
       REFERENCE
          "This document."
       INDEX { nhdpDiscIfSetRouterId }
    ::= { nhdpIib2HopSetPerfTable 1 }

    NhdpIib2HopSetPerfEntry ::=
       SEQUENCE {
          nhdpIib2HopSetPerfChanges
            Counter32,
          nhdpIib2HopSetPerfUpTime
            TimeTicks
         }

   nhdpIib2HopSetPerfChanges  OBJECT-TYPE
      SYNTAX      Counter32
```

```
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "This counter increments each
         time this 2-hop neighbor changes its
         N2_neighbor_iface_addr_list in the
         nhdpIib2HopSetTable."
      REFERENCE
         "The NHDP draft."
   ::= { nhdpIib2HopSetPerfEntry 1 }


   nhdpIib2HopSetPerfUpTime  OBJECT-TYPE
      SYNTAX      TimeTicks
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "This object returns the time in hundredths of
         a second when the 2-Hop Tuple
         corresponding to the given 2-hop neighbor IP address
         was registered in the nhdpIib2HopSetTable."
      REFERENCE
         "This document."
   ::= { nhdpIib2HopSetPerfEntry 2 }





--
-- nhdpNotifications
--

nhdpNotificationsControl OBJECT IDENTIFIER ::= { nhdpNotifications 1 }
nhdpNotificationsObjects OBJECT IDENTIFIER ::= { nhdpNotifications 2 }
nhdpNotificationsStates  OBJECT IDENTIFIER ::= { nhdpNotifications 3 }


   -- nhdpNotificationsControl

   nhdpSetNotification OBJECT-TYPE
          SYNTAX       OCTET STRING (SIZE(4))
          MAX-ACCESS   read-write
          STATUS       current
          DESCRIPTION
             "A 4-octet string serving as a bit map for
             the notification events defined by the NHDP
             notifications. This object is used to enable
```

```
             and disable specific NHDP notifications where
             a 1 in the bit field represents enabled. The
             right-most bit (least significant) represents
             notification 0.

             This object is persistent and when written
             the entity SHOULD save the change to
             non-volatile storage.
             "
           ::= { nhdpNotificationsControl 1 }

   nhdpNbrStateChangeThreshold OBJECT-TYPE
          SYNTAX       Integer32 (0..255)
          MAX-ACCESS   read-write
          STATUS       current
          DESCRIPTION
             "A threshold value for the
              nhdpNbrStateChange object.  If the
              number of occurences exceeds this threshold
              within the previous nhdpNbrStateChangeWindow,
              then the nhdpNbrStateChange notification
              is to be sent.
             "
           ::= { nhdpNotificationsControl 2 }

   nhdpNbrStateChangeWindow OBJECT-TYPE
          SYNTAX       TimeTicks
          MAX-ACCESS   read-write
          STATUS       current
          DESCRIPTION
             "A time window for the
              nhdpNbrStateChange object.  If the
              number of occurences exceeds the
              nhdpNbrStateChangeThreshold
              within the previous nhdpNbrStateChangeWindow,
              then the nhdpNbrStateChange notification
              is to be sent.

              This object represents the time in hundredths
              of a second.
             "
           ::= { nhdpNotificationsControl 3 }

   nhdp2HopNbrStateChangeThreshold OBJECT-TYPE
          SYNTAX       Integer32 (0..255)
          MAX-ACCESS   read-write
          STATUS       current
          DESCRIPTION
```

```
            "A threshold value for the
             nhdp2HopNbrStateChange object.  If the
             number of occurences exceeds this threshold
             within the previous nhdp2HopNbrStateChangeWindow,
             then the nhdp2HopNbrStateChange notification
             is to be sent.
            "
          ::= { nhdpNotificationsControl 4 }

   nhdp2HopNbrStateChangeWindow OBJECT-TYPE
         SYNTAX       TimeTicks
         MAX-ACCESS   read-write
         STATUS       current
         DESCRIPTION
            "A time window for the
             nhdp2HopNbrStateChange object.  If the
             number of occurences exceeds the
             nhdp2HopNbrStateChangeThreshold
             within the previous nhdp2HopNbrStateChangeWindow,
             then the nhdp2HopNbrStateChange notification
             is to be sent.

             This object represents the time in hundredths
             of a second.
            "
          ::= { nhdpNotificationsControl 5 }

   nhdpIfRxBadPacketThreshold OBJECT-TYPE
         SYNTAX       Integer32 (0..255)
         MAX-ACCESS   read-write
         STATUS       current
         DESCRIPTION
            "A threshold value for the
             nhdpIfRxBadPacket object.  If the
             number of occurences exceeds this threshold
             within the previous nhdpIfRxBadPacketWindow,
             then the nhdpIfRxBadPacket notification
             is to be sent.
            "
          ::= { nhdpNotificationsControl 6 }

   nhdpIfRxBadPacketWindow OBJECT-TYPE
         SYNTAX       TimeTicks
         MAX-ACCESS   read-write
         STATUS       current
         DESCRIPTION
            "A time window for the
             nhdpIfRxBadPacket object.  If the
```

```
              number of occurences exceeds the
              nhdpIfRxBadPacketThreshold
              within the previous nhdpIfRxBadPacketWindow,
              then the nhdpIfRxBadPacket notification
              is to be sent.

              This object represents the time in hundredths
              of a second.
             "
          ::= { nhdpNotificationsControl 7 }


   -- nhdpNotificationsObjects

   nhdpNbrStateChange NOTIFICATION-TYPE
          OBJECTS { nhdpDiscIfSetRouterId, -- The originator of
                                           --     the notification.
                    nhdpNbrState           -- The new state
                  }
          STATUS       current
          DESCRIPTION
             "nhdpNbrStateChange is a notification sent when a
             significant number of neighbors change their status
             (i.e. down, asymmetric, or symmetric) in a short
             time. The network administrator should select
             appropriate values for 'significant number of
             neighbors' and 'short time'."
          ::= { nhdpNotificationsObjects 1 }

    nhdp2HopNbrStateChange NOTIFICATION-TYPE
          OBJECTS { nhdpIib2HopSetIpAddress, -- The originator
                                             -- of the notification
                    nhdp2HopNbrState  -- The new state
             }
          STATUS       current
          DESCRIPTION
             "nhdp2HopNbrStateChange is a notification sent
             when a significant number of 2-hop neighbors
             change their status (i.e. up or down) in a short
             time. The network administrator should select
             appropriate values for 'significant number of
             neighbors' and 'short time'."
          ::= { nhdpNotificationsObjects 2 }

   nhdpIfRxBadPacket NOTIFICATION-TYPE
          OBJECTS { nhdpDiscIfSetRouterId, -- The originator of
                                           -- the notification
                    nhdpDiscIfSetIndex,  -- The interface on which the
```

```
                                  -- packet has been received
                    nhdpPacketSrcType,  -- The type of the source IP
                                      -- address of the packet
                    nhdpPacketSrc  -- The source IP address of
                                  -- the packet
             }
          STATUS       current
          DESCRIPTION
             "nhdpIfRxBadPacket is a notification sent when a
             significant number of incoming packets have not
             been successfully parsed in a short time. The
             network administrator should select appropriate
             values for 'significant number of neighbors'
             and 'short time'."
          ::= { nhdpNotificationsObjects 3 }


   nhdpIfStateChange NOTIFICATION-TYPE
          OBJECTS { nhdpIfIndex, -- The local interface
                    nhdpIfState  -- The new state
             }
          STATUS       current
          DESCRIPTION
             "nhdpIfStateChange is a notification sent when
             the status of an interface of this router has
             changed (i.e. an IP address has been added or
             removed to the interface, or the interface has
             changed its status from up to down or vice versa)."
          ::= { nhdpNotificationsObjects 4 }



    -- nhdpNotificationStates

    nhdpNbrState OBJECT-TYPE
       SYNTAX       INTEGER {
                       down (0),
                       asymmetric (1),
                       symmetric(2)
                       }
       MAX-ACCESS   read-only
       STATUS       current
       DESCRIPTION
          "NHDP neighbor states."
       DEFVAL { down }
       ::= { nhdpNotificationsStates 1 }
```

```
   nhdp2HopNbrState OBJECT-TYPE
       SYNTAX       INTEGER {
                       down (0),
                       up (1)
                       }
       MAX-ACCESS   read-only
       STATUS       current
       DESCRIPTION
          "NHDP 2hop neighbor states."
       DEFVAL { down }
       ::= { nhdpNotificationsStates 2 }

   nhdpIfState OBJECT-TYPE
       SYNTAX       INTEGER {
                       down (0),
                       up (1),
                       addresschange(2) -- If a new address has been
                                        -- added or an address has
                                        -- been removed
                       }
       MAX-ACCESS   read-only
       STATUS       current
       DESCRIPTION
          "NHDP interface states."
       DEFVAL { down }
       ::= { nhdpNotificationsStates 3 }

  nhdpPacketSrcType OBJECT-TYPE
           SYNTAX InetAddressType
           MAX-ACCESS read-only
           STATUS current
           DESCRIPTION
           "The IP address type of the
           address of an inbound packet that
           cannot be identified by a neighbor instance."
           ::= { nhdpNotificationsStates 4 }

   nhdpPacketSrc OBJECT-TYPE
          SYNTAX       InetAddress
          MAX-ACCESS   read-only
          STATUS       current
          DESCRIPTION
             "The IP address of an inbound packet that
             cannot be identified by a neighbor instance. When
             the last value of a notification using this object is
             needed, but no notifications of that type have been sent,
             this value pertaining to this object should
             be returned as 0.0.0.0 or :: respectively."
```

```
          ::= { nhdpNotificationsStates 5 }



--
-- nhdpConformance information
--

nhdpCompliances       OBJECT IDENTIFIER ::= { nhdpConformance 1 }
nhdpMIBGroups         OBJECT IDENTIFIER ::= { nhdpConformance 2 }


-- Compliance Statements
nhdpBasicCompliance MODULE-COMPLIANCE
  STATUS current
  DESCRIPTION
    "The basic implementation requirements for
     managed network entities that implement
     NHDP."
  MODULE -- this module

  MANDATORY-GROUPS { nhdpConfigurationGroup }


  ::= { nhdpCompliances 1 }

nhdpFullCompliance MODULE-COMPLIANCE
  STATUS current
  DESCRIPTION
    "The full implementation requirements for
     managed network entities that implement
     NHDP."
  MODULE -- this module

  MANDATORY-GROUPS { nhdpConfigurationGroup,
                     nhdpStateGroup,
                     nhdpPerformanceGroup,
                     nhdpNotificationObjectGroup,
                     nhdpNotificationGroup,
                     nhdpPerformanceGroup }

  ::= { nhdpCompliances 2 }


--
-- Units of Conformance
--
```

```
  nhdpConfigurationGroup OBJECT-GROUP
     OBJECTS {
              nhdpIfIndex,
              nhdpIfStatus,
              nhdpHelloInterval,
              nhdpHelloMinInterval,
              nhdpRefreshInterval,
              nhdpLHoldTime,
              nhdpHHoldTime,
              nhdpHystAcceptQuality,
              nhdpHystRejectQuality,
              nhdpInitialQuality,
              nhdpInitialPending,
              nhdpHpMaxJitter,
              nhdpHtMaxJitter,
              nhdpNHoldTime,
              nhdpIHoldTime,
              nhdpIfRowStatus
             }
      STATUS  current
      DESCRIPTION
         "Set of NHDP configuration objects implemented
          in this module."
   ::= { nhdpMIBGroups 2 }

  nhdpStateGroup OBJECT-GROUP
     OBJECTS {
              nhdpUpTime,
              nhdpDiscIfSetRouterId,
              nhdpDiscIfSetIndex,
              nhdpDiscIfSetIpAddrPrefixLen,
              nhdpLibRemovedIfAddrSetIpAddrPrefixLen,
              nhdpLibRemovedIfAddrSetIfIndex,
              nhdpLibRemovedIfAddrSetIrTime,
              nhdpIibLinkSetIfIndex,
              nhdpIibLinkSetLHeardTime,
              nhdpIibLinkSetLSymTime,
              nhdpIibLinkSetLPending,
              nhdpIibLinkSetLLost,
              nhdpIibLinkSetLTime,
              nhdpIib2HopSetIpAddressType,
              nhdpIib2HopSetIpAddress,
              nhdpIib2HopSet1HopIfIndex,
              nhdpIib2HopSetN2Time,
              nhdpNibNeighborSetNSymmetric,
              nhdpNibLostNeighborSetNLTime
             }
      STATUS  current
```

```
      DESCRIPTION
         "Set of NHDP state objects implemented
          in this module."
   ::= { nhdpMIBGroups 3 }

  nhdpPerformanceGroup OBJECT-GROUP
     OBJECTS {
              nhdpIfHelloMessageXmits,
              nhdpIfHelloMessageRecvd,
              nhdpIfHelloMessageXmitAccumulatedSize,
              nhdpIfHelloMessageRecvdAccumulatedSize,
              nhdpIfHelloMessageTriggeredXmits,
              nhdpIfHelloMessagePeriodicXmits,
              nhdpIfHelloMessageXmitAccumulatedSymmetricNeighborCount,
              nhdpIfHelloMessageXmitAccumulatedHeardNeighborCount,
              nhdpIfHelloMessageXmitAccumulatedLostNeighborCount,
              nhdpDiscIfRecvdPackets,
              nhdpDiscIfExpectedPackets,
              nhdpNibNeighborSetChanges,
              nhdpDiscNeighborNibNeighborSetChanges,
              nhdpDiscNeighborNibNeighborSetUpTime,
              nhdpDiscNeighborNibNeighborSetReachableLinkChanges,
              nhdpIib2HopSetPerfChanges,
              nhdpIib2HopSetPerfUpTime
            }
      STATUS  current
      DESCRIPTION
         "Set of NHDP performance objects implemented
          in this module."
   ::= { nhdpMIBGroups 4 }

    nhdpNotificationObjectGroup OBJECT-GROUP
      OBJECTS {
            nhdpSetNotification,
            nhdpNbrStateChangeThreshold,
            nhdpNbrStateChangeWindow,
            nhdp2HopNbrStateChangeThreshold,
            nhdp2HopNbrStateChangeWindow,
            nhdpIfRxBadPacketThreshold,
            nhdpIfRxBadPacketWindow,
            nhdpIfState,
            nhdpNbrState,
            nhdp2HopNbrState,
            nhdpPacketSrcType,
            nhdpPacketSrc
      }
      STATUS current
      DESCRIPTION
```

```
      "Set of NHDP notification objects implemented
      in this module."
   ::= { nhdpMIBGroups 5 }



  nhdpNotificationGroup NOTIFICATION-GROUP
     NOTIFICATIONS {
             nhdpNbrStateChange,
             nhdp2HopNbrStateChange,
             nhdpIfRxBadPacket,
             nhdpIfStateChange
             }
      STATUS  current
      DESCRIPTION
         "Set of NHDP notifications implemented
          in this module."
   ::= { nhdpMIBGroups 6 }


END
```

8. Security Considerations

This MIB defines objects for the configuration, monitoring and notification of the Neighborhood Discovery Protocol [NHDP]. NHDP allows routers to acquire topological information up to two hops away by virtue of exchanging HELLO messages. The information acquired by NHDP may be used by routing protocols. The neighborhood information, exchanged between routers using NHDP, serves these routing protocols as a baseline for calculating paths to all destinations in the MANET, relay set selection for network-wide transmissions etc.

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 can have a negative effect on network operations. These are the tables and objects and their sensitivity/vulnerability:

o nhdpIfStatus - this writable object turns on or off the NHDP process for the specified interface. If disabled, higher level protocol functions, e.g., routing, would fail causing network-wide disruptions.

o nhdpHelloInterval, nhdpHelloMinInterval, and nhdpRefreshInterval - these writable objects control the rate at which HELLO messages are sent on a wireless interface. If set at too high a rate, this could represent a form of DOS attack by overloading interface resources.

- nhdpHystAcceptQuality, nhdpHystRejectQuality, nhdpInitialQuality, nhdpInitialPending - these writable objects affect the perceived quality of the NHDP links and hence the overall stability of the network. If improperly set, these settings could result in network-wide disruptions.

- nhdpInterfaceTable - this table contains writable objects that affect the overall performance and stability of the NHDP process. Failure of the NHDP process would result in network-wide failure. Particularly sensitive objects from this table are discussed in the previous list items. This is the only table in the NHDP-MIB with writable objects.

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:

- nhdpDiscIfSetTable - The contains information on discovered neighbors, specifically their IP address in the nhdpDiscIfSetIpAddr object. This information provides an adversary broad information on the members of the MANET, located within this single table. This information can be use to expedite attacks on the other members of the MANET without having to go through a laborious discovery process on their own. This object is the index into the table, and has a MAX-ACCESS of 'not-accessible'. However, this information can be exposed using SNMP operations.

MANET technology is often deployed to support communications of emergency services or military tactical applications. In these applications, it is imperative to maintain the proper operation of the communications network and to protect sensitive information related to its operation. Therefore, when implementing these capabilities, the full use of SNMPv3 cryptographic mechanisms for authentication and privacy is RECOMMENDED.

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.

It is RECOMMENDED that implementers consider the security features as provided by the SNMPv3 framework (see [RFC3410], section 8), including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

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.

9. IANA Considerations

Editor's Note (to be removed prior to publication): the IANA is requested to assign a value for "XXXX" under the 'mib-2' subtree and to record the assignment in the SMI Numbers registry. When the assignment has been made, the RFC Editor is asked to replace "XXXX" (here and in the MIB module) with the assigned value and to remove this note. Note well: prior to official assignment by the IANA, a draft document MUST use placeholders (such as "XXXX" above) rather than actual numbers. See RFC4181 Section 4.5 for an example of how this is done in a draft MIB module.

10. Contributors

This MIB document uses the template authored by D. Harrington which is based on contributions from the MIB Doctors, especially Juergen Schoenwaelder, Dave Perkins, C.M.Heard and Randy Presuhn.

11. References

11.1. Normative References

[RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.

[RFC3418] Presuhn, R., "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3418, December 2002.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J.

Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

[NHDP] Clausen, T., Dearlove, C., and J. Dean, "Mobile Ad Hoc Network (MANET) Neighborhood Discovery Protocol (NHDP)", draft-ietf-manet-nhdp-15 (work in progress), December 2010.

[RFC4001] Daniele, M., Haberman, B., Routhier, S., and J. Schoenwaelder, "Textual Conventions for Internet Network Addresses", RFC 4001, February 2005.

11.2. Informative References

[REPORT] Cole, R., Macker, J., and A. Morton, "Definition of Managed Objects for Performance Reporting", draft-ietf-manet-report-mib-00 (work in progress), July 2010.

[RFC4750] Joyal, D., Galecki, P., Giacalone, S., Coltun, R., and F. Baker, "OSPF Version 2 Management Information Base", RFC 4750, December 2006.

[RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

[RFC3781] Strauss, F. and J. Schoenwaelder, "Next Generation Structure of Management Information (SMIng) Mappings to the Simple Network Management Protocol (SNMP)", RFC 3781, May 2004.

Appendix A. Open Issues

This section contains the set of open issues related to the development and design of the NHDP-MIB. This section will not be present in the final version of the MIB and will be removed once all the open issues have been resolved.

1. Check out the definitions of the Notification Group and their relationship within the subtree of the NHDP-MIB. Should we specify thresholds for neighbor change Notifications? How do we specify these?

2. Also, specify specific SNMP response to the snmp set request, i.e., 'generic error', 'bad value', etc.

Appendix B.

```
***************************************************************
* Note to the RFC Editor (to be removed prior to publication) *
*                                                             *
* 1) The reference to RFCXXXX within the DESCRIPTION clauses  *
* of the MIB module point to this draft and are to be         *
* assigned by the RFC Editor.                                 *
*                                                             *
* 2) The reference to RFCXXX2 throughout this document point  *
* to the current draft-ietf-manet-nhdp-mib-xx.txt.  This      *
* need to be replaced with the XXX RFC number.                *
*                                                             *
***************************************************************
```

Authors' Addresses

Ulrich Herberg
LIX, Ecole Polytechnique
Palaiseau Cedex, 91128
France

EMail: ulrich@herberg.name
URI: http://www.herberg.name/

Robert G. Cole
US Army CERDEC
328 Hopkins Road, Bldg 245
Aberdeen Proving Ground, Maryland 21005
USA

Phone: +1 410 278 6779
EMail: robert.g.cole@us.army.mil
URI: http://www.cs.jhu.edu/~rgcole/

Ian D Chakeres
CenGen
9250 Bendix Road North
Columbia, Maryland 560093
USA

EMail: ian.chakeres@gmail.com
URI: http://www.ianchak.com/

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: July 6, 2011

U. Herberg
LIX, Ecole Polytechnique
R. Cole
US Army CERDEC
T. Clausen
LIX, Ecole Polytechnique

January 2, 2011

Definition of Managed Objects for the Optimized Link State Routing Protocol version 2

draft-ietf-manet-olsrv2-mib-03

Abstract

This memo defines the Management Information Base (MIB) for configuring and managing the Optimized Link State Routing protocol version 2 (OLSRv2). The Optimized Link State Routing MIB is structured into state information, performance metrics, and notifications. This additional state and performance information is useful to troubleshoot problems and performance issues of the routing protocol. Different levels of compliance allow implementers to use smaller subsets of all defined objects, allowing for this MIB to be deployed on more constrained routers.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This memo defines the Management Information Base (MIB) for configuring and managing the Optimized Link State Routing protocol version 2 (OLSRv2). The Optimized Link State Routing MIB is structured into state information, performance metrics, and notifications. In addition to configuration, this additional state and performance information is useful to troubleshoot problems and performance issues of the routing protocol. Different levels of compliance allow implementers to use smaller subsets of all defined objects, allowing for this MIB to be deployed on more constrained routers.

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 [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 [RFC2578], [RFC2579], and [RFC2580].

3. Conventions

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 [RFC2119].

4. Overview

The Optimized Link State Routing Protocol version 2 (OLSRv2) [OLSRv2] is a table driven, proactive routing protocol, i.e. it exchanges topology information with other routers in the network regularly. OLSRv2 is an optimization of the classical link state routing protocol. Its key concept is that of MultiPoint Relays (MPRs). Each router selects a set of its neighbor routers (which "cover" all of its symmetrically connected 2-hop neighbor routers) as MPRs. MPRs are then used to achieve both flooding reduction and topology reduction.

This MIB document provides management and control capabilities of an OLSRv2 instance, allowing to monitor the state and performance of an

OLSRV2 router, as well as to change settings of the deployment.

As OLSRv2 relies on the neighborhood information discovered by NHDP [NHDP], the OLSRv2-MIB is aligned with the NHDP-MIB [NHDP-MIB]. In particular, common indexes for router interfaces and discovered neighbors are used, as described in Section 5.2.

4.1. Terms

The following definitions apply throughout this document:

o Configuration Objects - switches, tables, objects which are initialized to default settings or set through the management interface defined by this MIB.

o State Objects - automatically generated values which define the current operating state of the OLSRv2 protocol process in the router.

o Performance Objects - automatically generated values which help an administrator or automated tool to assess the performance of the OLSRv2 routing process on the router.

o Notification Objects - define triggers and associated notification messages allowing for asynchronous tracking of pre-defined events on the managed router.

5. Structure of the MIB Module

This section presents the structure of the OLSRv2-MIB module. The objects are arranged into the following structure:

o olsrv2Objects - defines objects forming the basis for the OLSRv2-MIB. These objects are divided up by function into the following groups:

  * Configuration Group - defining objects related to the configuration of the OLSRv2 instance on the router.

  * State Group - defining objects which reflect the current state of the OLSRv2 instance running on the router.

  * Performance Group -defining objects which are useful to a management station when characterizing the performance of OLSRv2 on the router and in the MANET.

- o olsrv2Notifications - objects defining OLSRv2-MIB notifications.
- o olsrv2Conformance - defining the minimal and maximal conformance requirements for implementations of this MIB.

5.1. The Configuration Group

The OLSRv2 router is configured with a set of controls. The authoritative list of configuration controls within the OLSRv2-MIB are found within the MIB module itself. Generally, an attempt was made in developing the OLSRv2-MIB module to support all configuration objects defined in [OLSRv2]. For all of the configuration parameters, the same constraints and default values of these parameters as defined in [OLSRv2] are followed.

5.2. The State Group

The State Group reports current state information of a router running [OLSRv2]. The OLSRv2-MIB State Group tables were designed to contain the complete set of state information defined within the information bases in [OLSRv2].

The OLSRv2-MIB State Group tables are constructed as extensions to the corresponding tables within the State Group of the NHDP-MIB [NHDP-MIB]. Further, the State Group tables defined in this MIB are aligned with the according tables in the NHDP-MIB [NHDP-MIB], as described in Section 6.2.

5.3. The Performance Group

The Performance Group reports values relevant to system performance. This section lists objects for OLSRv2 performance monitoring, some of which explicitly appear in the OLSRv2-MIB and others which are obtainable through a combination of base objects from this MIB and reports available through the REPORT-MIB [REPORT]. Throughout this section, those objects will be pointed out that are intended as base objects, which are explicitly defined within the OLSRv2-MIB and those objects which are derived through a combination of the base objects within the OLSRv2-MIB and capabilities afforded by the REPORT-MIB.

The objects in this group can be used to examine stability of the Routing Set, the selected MPRs, as well as message scheduling of this router.

5.3.1. Recalculation Performance Objects

The following objects return statistics to the frequency of Routing Set recalculations.

- Number of Routing Set recalculations

  This object counts each recalculation of the Routing Set.

  This is a Base Object.

  Object name: olsrv2RoutingSetRecalculationCount

  Object type: Counter32

- Acquire history of Routing Set recalculations

  This object returns the history of the exact timestamps of each time the Routing Set has been recalculated.

  This is a Derived Object to be pulled from the REPORT-MIB. It is derived from, e.g., the olsrv2RoutingSetRecalculationCount Base Object from the OLSRv2-MIB along with the capabilities derived from the reportHistoryGroup from the REPORT-MIB.

- Histogram of the intervals between Routing Set recalculations

  Returns the values that represent a histogram of intervals between Routing Set recalculations.

  This is a Derived Object to be pulled from the REPORT-MIB. It can be derived from, e.g., the olsrv2RoutingSetRecalculationCount Base Object from the OLSRv2-MIB along with the capabilities derived from the reportHistoryGroup from the REPORT-MIB. The network management application could convert this information into the desired histogram.

- Changes of the frequency of the Routing Set recalculations

  This object will divide the given time interval from t0 to t1 into a given number of equal parts. It then creates a histogram for each part and calculate the distances (using the Bhattacharyya distance) between each two adjacent histograms in time. A higher value between two histograms means more difference between the histograms.

  This is a Derived Object to be pulled from the REPORT-MIB, as previously discussed, albeit this is a bit more complex with respect to the management application.

The following objects return statistics to the frequency of recalculating the MPRs of this router.

- Number of MPR recalculations

  This object counts each recalculation of the MPRs of the router.

  This is a Base Object.

  Object name: olsrv2MPRSetRecalculationCount

  Object type: Counter32

- Acquire history of MPR recalculations

  This object returns the history of the exact timestamps of each time the MPRs have been recalculated.

  This is a Derived Object to be pulled from the REPORT-MIB. It is derived from, e.g., the olsrv2MPRSetRecalculationCount Base Object from the OLSRv2-MIB along with the capabilities derived from the reportHistoryGroup from the REPORT-MIB.

- Histogram of the intervals between MPR recalculations

  Returns the values that represent a histogram of intervals between MPR recalculations. The histogram includes all changes that have been made after the given time t0 and before the given time t1.

  This is a Derived Object to be pulled from the REPORT-MIB. It can be derived from, e.g., the olsrv2MPRSetRecalculationCount Base Object from the OLSRv2-MIB along with the capabilities derived from the reportHistoryGroup from the REPORT-MIB. The network management application could convert this information into the desired histogram.

- Changes of the frequency of MPR recalculations

  This object will divide the given time interval from t0 to t1 into a given number of equal parts. It then creates a histogram for each part and calculate the distances (using the Bhattacharyya distance) between each two adjacent histograms in time. A higher value between two histograms means more difference between the histograms.

  This is a Derived Object to be pulled from the REPORT-MIB, as previously discussed, albeit this is a bit more complex with respect to the management application.

5.3.2. Message-related Performance Objects

The following objects return some of the statistics related to TC messages:

- Total number of sent TC messages on an interface

  This is a Base Object.

  Object name: olsrv2IfTcMessageXmits

  Object type: Counter32

- Total number of received TC messages on an interface

  This is a Base Object.

  Object name: olsrv2IfTcMessageRecvd

  Object type: Counter32

- Total number of sent periodic TC messages on an interface

  This is a Base Object.

  Object name: olsrv2IfTcMessagePeriodicXmits

  Object type: Counter32

- Total number of sent triggered TC messages on an interface

  This is a Base Object.

  Object name: olsrv2IfTcMessageTriggeredXmits

  Object type: Counter32

- Total number of forwarded TC messages on an interface

  This is a Base Object.

  Object name: olsrv2IfTcMessageForwardedXmits

  Object type: Counter32

- Acquire history of TC message scheduling instance for the given time duration on an interface

This object returns the history of the exact timestamps of each TC message that has been sent as well as the type of the message (triggered or periodical). The list of events starts at the given point of time t0 and ends at the given time t1.

This is a Derived Object to be pulled from the REPORT-MIB. It is derived from, e.g., the olsrv2IfTcMessagePeriodicXmits and olsrv2IfTcMessageTriggeredXmits Base Objects from the OLSRv2-MIB along with the capabilities derived from the reportHistoryGroup from the REPORT-MIB.

- Histogram of the intervals between TC messages on an interface

  Returns the values (in a 2-dimensional array) that represent a histogram of intervals between TC messages, separated by periodic and triggered TC. The histogram displays the distribution of intervals between two consecutive TC of the same type (triggered or periodical) using a given bin size. It includes all TC that have been sent after the given time t0 and before the given time t1.

  This is a Derived Object to be pulled from the REPORT-MIB. It can be derived from, e.g., the olsrv2IfTcMessagePeriodicXmits and olsrv2IfTcMessageTriggeredXmits Base Objects from the OLSRv2-MIB along with the capabilities derived from the reportHistoryGroup from the REPORT-MIB. The network management application could convert this information into the desired histogram.

- Changes of the frequency of the message scheduling on an interface

  This object will divide the given time interval from t0 to t1 into a given number of equal parts. It then creates a histogram for each part and calculate the distances (using the Bhattacharyya distance) between each two adjacent histograms in time. A higher value between two histograms means more difference between the histograms. For instance, that could happen if suddenly many triggered TC messages are sent, whereas before there have been only very few such triggered messages.

  This is a Derived Object to be pulled from the REPORT-MIB, as previously discussed, albeit this is a bit more complex with respect to the management application.

- Average number of sent TC messages per second between the given time t0 and t1 on an interface

This is a Derived Object to be pulled from the reportSampledGroup from the REPORT-MIB. It is derived from, e.g., the olsrv2IfTcMessageXmits Base Object.

- Average number of received TC messages per second between the given time t0 and t1 on an interface

  This is a Derived Object to be pulled from the REPORT-MIB. See the previous discussion.

- Total accumulated size in octets of sent TC messages on an interface

  This is a Base Object.

  Object name: olsrv2IfHelloMessageXmitAccumulatedSize

  Object type: Counter32

- Total accumulated size in octets of received TC messages on an interface

  This is a Base Object.

  Object name: olsrv2IfHelloMessageRecvdAccumulatedSize

  Object type: Counter32

- Average size in octets of sent TC messages per second between the given time t0 and t1 on an interface

  This is a Derived Object to be pulled from the REPORT-MIB. See the previous discussion.

- Average size in octets of received TC messages per second between the given time t0 and t1 on an interface

  This is a Derived Object to be pulled from the REPORT-MIB. See the previous discussion.

- Total accumulated number of advertised MPR selectors in TC messages on an interface

  This is a Base Object.

Object name:
olsrv2IfHelloMessageXmitAccumulatedSymmetricNeighborCount

Object type: Counter32

5.4. The Notifications Group

The Notifications Subtree contains the list of notifications supported within the OLSRv2-MIB and their intended purpose or utility.

The same mechanisms for improving the network performance by reducing the number of notifications apply as defined in Section 5.4 of [NHDP-MIB]. The Notifications Group contains Control, Objects and States, where the Control contains definitions of objects to control the frequency of notifications being sent. The Objects define the supported notifications and the State is used to define additional information to be carried within the notifications.

6. Relationship to Other MIB Modules

This section specifies the relationship of the MIB modules contained in this document to other standards, particularly to standards containing other MIB modules. Definitions imported from other MIB modules and other MIB modules that SHOULD be implemented in conjunction with the MIB module contained within this document are identified in this section.

6.1. Relationship to the SNMPv2-MIB

The 'system' group in the SNMPv2-MIB [RFC3418] is defined as being mandatory for all systems, and the objects apply to the entity as a whole. The 'system' group provides identification of the management entity and certain other system-wide data. The OLSRv2-MIB does not duplicate those objects.

6.2. Relationship to the NHDP-MIB

OLSRv2 depends on the neighborhood information that is discovered by [NHDP]. In order access the Objects relating to discovered neighbors, the State Group tables of the NHDP-MIB [NHDP-MIB] are aligned with this MIB. This is accomplished through the definition of two TEXTUAL-CONVENTIONS in the NHDP-MIB: the NeighborInterfaceId and the NeighborRouterId. These object types are used to develop indexes into common NHDP-MIB and routing protocol State Group tables. These objects are locally significant but should be locally common to the NHDP-MIB and the OLSRv2-MIB implemented on a common networked router. This will allow for improved cross referencing of information across the two MIBs.

6.3. Relationship to the REPORT-MIB

This document describes several Performance Management metrics for the management of OLSRv2 routers. However, not all of these metrics are explicitly defined solely within the context of this OLSRv2-MIB. Some of these metrics are obtained through joint interaction between this MIB and the REPORT-MIB [REPORT]. This OLSRv2-MIB defines the minimum necessary objects (often of type COUNTER) which form the underlying basis for more sophisticated Performance Management reporting available in conjunction with the REPORT-MIB. See Section 5.3 for a description of the performance metrics for OLSRv2.

6.4. MIB modules required for IMPORTS

The following OLSRv2-MIB module IMPORTS objects from NHDP-MIB [NHDP-MIB], SNMPv2-SMI [RFC2578], SNMPv2-TC [RFC2579], SNMPv2-CONF [RFC2580], IF-MIB [RFC2863], INET-ADDRESS-MIB [RFC4001], and SMIng [RFC3781].

7. Definitions

This section contains the MANET-OLSRv2-MIB module defined by the specification.

```
MANET-OLSRv2-MIB DEFINITIONS ::= BEGIN

IMPORTS
   MODULE-IDENTITY, OBJECT-TYPE, Counter32,
   Integer32, Unsigned32, mib-2, TimeTicks,
   NOTIFICATION-TYPE
            FROM SNMPv2-SMI --[RFC2578]

   TimeStamp, TruthValue, RowStatus
            FROM SNMPv2-TC --[RFC2579]

   MODULE-COMPLIANCE, OBJECT-GROUP
            FROM SNMPv2-CONF --[STD58]

   InetAddressType, InetAddress,
   InetAddressPrefixLength
            FROM INET-ADDRESS-MIB --[RFC3291]

   InterfaceIndexOrZero
            FROM IF-MIB --[RFC2863]
```

```
      Float32TC
               FROM FLOAT-TC-MIB --[RFCXXXX]

      NeighborRouterId
               FROM NHDP-MIB -- [draft-ietf-manet-nhdp-mib]
      ;

   manetOlsrv2MIB MODULE-IDENTITY
      LAST-UPDATED "201101021000Z"   -- January 02, 2011
      ORGANIZATION "IETF MANET Working Group"
      CONTACT-INFO
         "WG E-Mail: manet@ietf.org

          WG Chairs: ian.chakeres@gmail.com
                     jmacker@nrl.navy.mil


          Editors:   Ulrich Herberg
                     Ecole Polytechnique
                     LIX
                     91128 Palaiseau Cedex
                     France
                     +33 1 69 33 41 26
                     ulrich@herberg.name
                     http://www.herberg.name/

                     Thomas Heide Clausen
                     Ecole Polytechnique
                     LIX
                     91128 Palaiseau Cedex
                     France
                     http://www.thomasclausen.org/
                     T.Clausen@computer.org

                     Robert G. Cole
                     US Army CERDEC
                     Space and Terrestrial Communications
                     328 Hopkins Road
                     Bldg 245, Room 16
                     Aberdeen Proving Ground, MD 21005
                     USA
                     +1 410 278-6779
                     robert.g.cole@us.army.mil
                     http://www.cs.jhu.edu/~rgcole/"

      DESCRIPTION
         "This MIB module contains managed object definitions
          for the Manet OLSRv2 routing process defined in the
```

```
          Optimized Link State Routing Protocol version 2
          defined in [RFCXXXX].

          Copyright (C) The IETF Trust (2009). This version
          of this MIB module is part of RFC xxxx; see the RFC
          itself for full legal notices."

        -- Revision History
        REVISION    "201101021000Z"   -- Jan 02, 2011
        DESCRIPTION
         "The sixth version of this MIB module,
          published as draft-ietf-manet-olsrv2-mib-03.txt.
          Changes made in this version include
          the addition of the NotificationGroup,
          updates to the ConformanceGroup and
          fixes discovered from running smilint.
          Finally, added the olsrv2OrigIpAddrType and
          olsrv2OrigIpAddr objects to the
          Configuration Group to identify
          this OLSRv2 router."
        REVISION    "201007121000Z"   -- July 12, 2010
        DESCRIPTION
         "The fifth version of this MIB module,
          published as draft-ietf-manet-olsrv2-mib-02.txt.
          Many editorial changes, Security Considerations,
          corrected errors in the MIB."
        REVISION    "200911091000Z"   -- Nov 9, 2009
        DESCRIPTION
         "The fourth version of this MIB module,
          published as draft-ietf-manet-olsrv2-mib-01.txt.
          Added Performance objects, and updated to newest
          OLSRv2 draft."
        REVISION    "200905031300Z"   -- May 3, 2009
        DESCRIPTION
           "Third draft of this MIB module published as
            draft-ietf-manet-olsrv2-mib-00.txt. Rev'd
            as a new MANET WG document.  Cleaned up SYNTAX
            errors and other typos found by 'smilint'."
        REVISION    "200902151300Z"   -- February 15, 2009
        DESCRIPTION
           "Second draft of this MIB module published as
            draft-cole-manet-olsrv2-mib-01.txt.  Cleaned up
            table indexing and aligned with the NHDP-MIB
            draft (draft-cole-manet-nhdp-mib-01.txt)."
        REVISION    "200810241300Z"   -- October 24, 2008
        DESCRIPTION
           "Initial draft of this MIB module published as
            draft-cole-manet-olsrv2-mib-00.txt."
```

```
        -- RFC-Editor assigns XXXX
        ::= { mib-2 998 }   -- to be assigned by IANA


   --
   -- TEXTUAL CONVENTIONs
   --

   -- none


   --
   -- Top-Level Object Identifier Assignments
   --

   olsrv2MIBNotifications OBJECT IDENTIFIER ::= { manetOlsrv2MIB 0 }
   olsrv2MIBObjects       OBJECT IDENTIFIER ::= { manetOlsrv2MIB 1 }
   olsrv2MIBConformance   OBJECT IDENTIFIER ::= { manetOlsrv2MIB 2 }

   --
   -- olsrv2ConfigurationGroup
   --
   --    Contains the OLSRv2 objects that configure specific
   --    options that determine the overall performance and operation
   --    of the OLSRv2 routing process.
   --

   olsrv2ConfigurationGroup OBJECT IDENTIFIER ::= {olsrv2MIBObjects 1}


    olsrv2OrigIpAddrType  OBJECT-TYPE
       SYNTAX      InetAddressType
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "The type of the olsrv2OrigIpAddr, as defined
           in the InetAddress MIB [RFC 4001].

           This object is persistent and when written
           the entity SHOULD save the change to
           non-volatile storage.
          "
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2ConfigurationGroup 1 }

    olsrv2OrigIpAddr  OBJECT-TYPE
       SYNTAX      InetAddress
```

```
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "An address which is unique (within the MANET)
           to a router.  A router MUST select an
           originator address; it MAY choose one of
           its interface addresses as its originator
           address.  If it selects a routable address
           then this MUST be one which this router will
           accept as destination.  An originator address
           MUST NOT have a prefix length, except for
           when included in an Address Block where it MAY
           be associated with a prefix of maximum prefix
           length (e.g., if the originator address is an
           IPv6 address, it MUST have either no prefix
           length, or have a prefix length of 128).
           An originator address may be a routable or
           non-routable address.

           This object is persistent and when written
           the entity SHOULD save the change to
           non-volatile storage.
       "
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2ConfigurationGroup 2 }

   --
   -- Local history times
   --

   olsrv2OHoldTime  OBJECT-TYPE
      SYNTAX      Unsigned32
      UNITS       "milliseconds"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "olsrv2OHoldTime corresponds to
         O_HOLD_TIME of OLSRv2.

         This object is persistent and when written
         the entity SHOULD save the change to
         non-volatile storage."
      REFERENCE
         "The OLSRv2 draft.
          Section 5 on Protocol Parameters."
      DEFVAL { 30000 }
   ::= { olsrv2ConfigurationGroup 3 }
```

```
   --
   -- Message intervals
   --

   olsrv2TcInterval  OBJECT-TYPE
      SYNTAX      Unsigned32
      UNITS       "milliseconds"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "olsrv2TcInterval corresponds to
         TC_INTERVAL of OLSRv2.

         The following constraints apply to this
         parameter:

             olsrv2TcInterval > 0
             olsrv2TcInterval &gt;= olsrv2TcMinInterval

         This object is persistent and when written
         the entity SHOULD save the change to
         non-volatile storage."
      REFERENCE
         "The OLSRv2 draft.
          Section 5 on Protocol Parameters."
      DEFVAL { 5000 }
   ::= { olsrv2ConfigurationGroup 4 }

   olsrv2TcMinInterval  OBJECT-TYPE
      SYNTAX      Unsigned32
      UNITS       "milliseconds"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "olsrv2TcMinInterval corresponds to
         TC_MIN_INTERVAL of OLSRv2.

         The following constraint applies to this
         parameter:

             olsrv2TcInterval &gt;= olsrv2TcMinInterval


         This object is persistent and when written
         the entity SHOULD save the change to
         non-volatile storage."
      REFERENCE
         "The OLSRv2 draft.
```

```
          Section 5 on Protocol Parameters."
      DEFVAL { 1250 }
   ::= { olsrv2ConfigurationGroup 5 }


   --
   -- Advertised information validity times
   --

   olsrv2THoldTime  OBJECT-TYPE
      SYNTAX      Unsigned32
      UNITS       "milliseconds"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "olsrv2THoldTime corresponds to
         T_HOLD_TIME of OLSRv2.

         The following constraint applies to this
         parameter:

             olsrv2THoldTime &gt;= olsrv2TcInterval

         If TC messages can be lost, then
         olsrv2THoldTime SHOULD be
         significantly greater than olsrv2TcInterval;
         a value &gt;= 3 x olsrv2TcInterval is RECOMMENDED.

         olsrv2THoldTime MUST be representable as
         described in [timetlv].

         This object is persistent and when written
         the entity SHOULD save the change to
         non-volatile storage."
      REFERENCE
         "The OLSRv2 draft.
          Section 5 on Protocol Parameters."
      DEFVAL { 15000 }
   ::= { olsrv2ConfigurationGroup 6 }

   olsrv2AHoldTime  OBJECT-TYPE
      SYNTAX      Unsigned32
      UNITS       "milliseconds"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "olsrv2AHoldTime corresponds to
         A_HOLD_TIME of OLSRv2.
```

```
         If TC messages can be lost, then
         olsrv2AHoldTime SHOULD be
         significantly greater than olsrv2TcInterval;
         a value &gt;= 3 x olsrv2TcInterval is
         RECOMMENDED.

         olsrv2AHoldTime MUST be representable as
         described in [timetlv].

         This object is persistent and when written
         the entity SHOULD save the change to
         non-volatile storage."
      REFERENCE
         "The OLSRv2 draft.
          Section 5 on Protocol Parameters."
      DEFVAL { 15000 }
   ::= { olsrv2ConfigurationGroup 7 }

   --
   -- Received message validity times
   --

   olsrv2RxHoldTime  OBJECT-TYPE
      SYNTAX      Unsigned32
      UNITS       "milliseconds"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "olsrv2RxHoldTime corresponds to
         RX_HOLD_TIME of OLSRv2.

         The following constraint applies to this
         parameter:

             olsrv2RxHoldTime > 0

         This parameter SHOULD be greater
         than the maximum difference in time that a
         message may take to traverse the MANET,
         taking into account any message forwarding
         jitter as well as propagation, queuing,
         and processing delays.

         This object is persistent and when written
         the entity SHOULD save the change to
         non-volatile storage."
      REFERENCE
         "The OLSRv2 draft.
```

```
           Section 5 on Protocol Parameters."
       DEFVAL { 30000 }
    ::= { olsrv2ConfigurationGroup 8 }

    olsrv2PHoldTime  OBJECT-TYPE
       SYNTAX      Unsigned32
       UNITS       "milliseconds"
       MAX-ACCESS  read-write
       STATUS      current
       DESCRIPTION
          "olsrv2PHoldTime corresponds to
          P_HOLD_TIME of OLSRv2.

          The following constraint applies to this
          parameter:

              olsrv2PHoldTime > 0

          This parameter SHOULD be greater
          than the maximum difference in time that a
          message may take to traverse the MANET,
          taking into account any message forwarding
          jitter as well as propagation, queuing,
          and processing delays.

          This object is persistent and when written
          the entity SHOULD save the change to
          non-volatile storage."
       REFERENCE
          "The OLSRv2 draft.
           Section 5 on Protocol Parameters."
       DEFVAL { 30000 }
    ::= { olsrv2ConfigurationGroup 9 }

    olsrv2FHoldTime  OBJECT-TYPE
       SYNTAX      Unsigned32
       UNITS       "milliseconds"
       MAX-ACCESS  read-write
       STATUS      current
       DESCRIPTION
          "olsrv2RxHoldTime corresponds to
          RX_HOLD_TIME of OLSRv2.

          The following constraint applies to this
          parameter:

              olsrv2FHoldTime > 0
```

```
         This parameter SHOULD be greater
         than the maximum difference in time that a
         message may take to traverse the MANET,
         taking into account any message forwarding
         jitter as well as propagation, queuing,
         and processing delays.

         This object is persistent and when written
         the entity SHOULD save the change to
         non-volatile storage."
      REFERENCE
         "The OLSRv2 draft.
          Section 5 on Protocol Parameters."
      DEFVAL { 30000 }
   ::= { olsrv2ConfigurationGroup 10 }


   --
   -- Jitter times
   --

   olsrv2TpMaxJitter  OBJECT-TYPE
      SYNTAX      Unsigned32
      UNITS       "milliseconds"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "olsrv2TpMaxJitter corresponds to
         TP_MAXJITTER of OLSRv2.

         This object is persistent and when written
         the entity SHOULD save the change to
         non-volatile storage."
      REFERENCE
         "The OLSRv2 draft.
          Section 5 on Protocol Parameters."
      DEFVAL { 500 }
   ::= { olsrv2ConfigurationGroup 11 }

   olsrv2TtMaxJitter  OBJECT-TYPE
      SYNTAX      Unsigned32
      UNITS       "milliseconds"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "olsrv2TtMaxJitter corresponds to
         TT_MAXJITTER of OLSRv2.
```

```
         This object is persistent and when written
         the entity SHOULD save the change to
         non-volatile storage."
      REFERENCE
         "The OLSRv2 draft.
          Section 5 on Protocol Parameters."
      DEFVAL { 500 }
   ::= { olsrv2ConfigurationGroup 12 }

   olsrv2FMaxJitter  OBJECT-TYPE
      SYNTAX      Unsigned32
      UNITS       "milliseconds"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "olsrv2FMaxJitter corresponds to
         F_MAXJITTER of OLSRv2.

         This object is persistent and when written
         the entity SHOULD save the change to
         non-volatile storage."
      REFERENCE
         "The OLSRv2 draft.
          Section 5 on Protocol Parameters."
      DEFVAL { 500 }
   ::= { olsrv2ConfigurationGroup 13 }


   --
   -- Hop limits
   --

   olsrv2TcHopLimit  OBJECT-TYPE
      SYNTAX      Unsigned32 (0..255)
      UNITS       "hops"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "olsrv2TcHopLimit corresponds to
         TC_HOP_LIMIT of OLSRv2.

         The following constraint applies to this
         parameter:

         The maximum value of
         olsrv2TcHopLimit &gt;= the network diameter
         in hops, a value of 255 is RECOMMENDED.
```

```
         All values of olsrv2TcHopLimit &gt;= 2.

         This object is persistent and when written
         the entity SHOULD save the change to
         non-volatile storage."
       REFERENCE
         "The OLSRv2 draft.
          Section 5 on Protocol Parameters."
       DEFVAL { 255 }
   ::= { olsrv2ConfigurationGroup 14 }


   --
   -- Willingness
   --

   olsrv2Willingness  OBJECT-TYPE
      SYNTAX      Unsigned32 (0..255)
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "olsrv2Willingness corresponds to
         WILLINGNESS of OLSRv2.

         The following constraint applies to this
         parameter:

         WILL_NEVER (0) &lt;= olsrv2Willingness &lt;=
                                 WILL_ALWAYS (15)

         This object is persistent and when written
         the entity SHOULD save the change to
         non-volatile storage."
      REFERENCE
         "The OLSRv2 draft.
          Section 5 on Protocol Parameters."
      DEFVAL { 7 }
   ::= { olsrv2ConfigurationGroup 15 }



   --
   -- olsrv2StateGroup
   --

   -- Contains information describing the current state of
   -- the OLSRv2 process.
```

```
   olsrv2StateGroup  OBJECT IDENTIFIER ::= { olsrv2MIBObjects 2 }

   olsrv2RouterStatus  OBJECT-TYPE
      SYNTAX      TruthValue
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The current status of the OLSRv2 router
          routing process."
   ::= { olsrv2StateGroup 1 }



    --
    -- Local Information Base  - as defined in [NHDP],
    -- extended by the addition of an Originator Set,
    -- defined in Section 6.1 and a Local Attached
    -- Network Set, defined in Section 6.2.
    --



    --
    -- Originator Set
    --

    olsrv2LibOrigSetTable OBJECT-TYPE
       SYNTAX       SEQUENCE OF Olsrv2LibOrigSetEntry
       MAX-ACCESS   not-accessible
       STATUS       obsolete
       DESCRIPTION
          "A router's Originator Set records addresses
          that were recently used as originator addresses
          by this router.  If a router's originator
          address is immutable then this set is always
          empty and MAY be omitted."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2StateGroup 2 }

    olsrv2LibOrigSetEntry  OBJECT-TYPE
       SYNTAX      Olsrv2LibOrigSetEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
          "A router's Originator Set consists of
          Originator Tuples:
            (O_orig_addr, O_time)."
```

```
       REFERENCE
          "The OLSRv2 draft."
       INDEX { olsrv2LibOrigSetIpAddr }
    ::= { olsrv2LibOrigSetTable 1 }

    Olsrv2LibOrigSetEntry ::=
       SEQUENCE {
          olsrv2LibOrigSetIpAddrType
            InetAddressType,
          olsrv2LibOrigSetIpAddr
            InetAddress,
          olsrv2LibOrigSetExpireTime
            TimeStamp
         }

    olsrv2LibOrigSetIpAddrType  OBJECT-TYPE
       SYNTAX      InetAddressType
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "The type of the olsrv2LibOrigSetIpAddr, as defined
           in the InetAddress MIB [RFC 4001]."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2LibOrigSetEntry 1 }

    olsrv2LibOrigSetIpAddr  OBJECT-TYPE
       SYNTAX      InetAddress
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "A recently used originator address
           by this router."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2LibOrigSetEntry 2 }

     olsrv2LibOrigSetExpireTime  OBJECT-TYPE
       SYNTAX      TimeStamp
       UNITS      "milliseconds"
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
          "This object specifies the time at which this
           entry expires and MUST be removed."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2LibOrigSetEntry 3 }
```

```
    --
    -- Local Attached Network Set
    --

    olsrv2LibLocAttNetSetTable OBJECT-TYPE
      SYNTAX       SEQUENCE OF Olsrv2LibLocAttNetSetEntry
      MAX-ACCESS   not-accessible
      STATUS       obsolete
      DESCRIPTION
         "A router's Local Attached Network Set records
         its local non-OLSRv2 interfaces via which it
         can act as gateways to other networks. The
         Local Attached Network Set is not modified by
         this protocol."
      REFERENCE
         "The OLSRv2 draft."
   ::= { olsrv2StateGroup 3 }

    olsrv2LibLocAttNetSetEntry  OBJECT-TYPE
       SYNTAX      Olsrv2LibLocAttNetSetEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
          "The entries include the Local Attached
           Network Tuples:

              (AL_net_addr, AL_dist)

           where:

              AL_net_addr is the network address
              of an attached network which can
              be reached via this router.

              AL_dist is the number of hops to
              the network with address AL_net_addr
              from this router."
       REFERENCE
          "The OLSRv2 draft."
       INDEX { olsrv2LibLocAttNetSetIpAddr,
               olsrv2LibLocAttNetSetIpAddrPrefixLen }
    ::= { olsrv2LibLocAttNetSetTable 1 }

    Olsrv2LibLocAttNetSetEntry ::=
       SEQUENCE {
          olsrv2LibLocAttNetSetIpAddrType
            InetAddressType,
          olsrv2LibLocAttNetSetIpAddr
```

```
            InetAddress,
          olsrv2LibLocAttNetSetIpAddrPrefixLen
            InetAddressPrefixLength,
          olsrv2LibLocAttNetSetDistance
            Unsigned32,
          olsrv2LibLocAttNetSetRowStatus
            RowStatus
         }

    olsrv2LibLocAttNetSetIpAddrType  OBJECT-TYPE
       SYNTAX      InetAddressType
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "The type of the olsrv2LibLocAttNetSetIpAddr, as defined
           in the InetAddress MIB [RFC 4001]."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2LibLocAttNetSetEntry 1 }

    olsrv2LibLocAttNetSetIpAddr  OBJECT-TYPE
       SYNTAX      InetAddress
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "This is the network address of an attached
           network which can be reached via this router."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2LibLocAttNetSetEntry 2 }

    olsrv2LibLocAttNetSetIpAddrPrefixLen  OBJECT-TYPE
      SYNTAX      InetAddressPrefixLength
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "Indicates the number of leading one bits that form the
          mask to be logical-ANDed with the destination address
          before being compared to the value in the
          olsrv2LibLocAttNetSetIpAddr field."
      REFERENCE
         "The OLSRv2 draft."
   ::= { olsrv2LibLocAttNetSetEntry 3 }

   olsrv2LibLocAttNetSetDistance  OBJECT-TYPE
      SYNTAX      Unsigned32 (1..255)
      UNITS      "hops"
      MAX-ACCESS  read-only
```

```
      STATUS      current
      DESCRIPTION
         "This object specifies the number of hops
          to the network with address
          olsrv2LibLocAttNetSetIpAddr from this router."
      REFERENCE
         "The OLSRv2 draft."
   ::= { olsrv2LibLocAttNetSetEntry 4 }

   olsrv2LibLocAttNetSetRowStatus  OBJECT-TYPE
      SYNTAX      RowStatus
      MAX-ACCESS  read-create
      STATUS      current
      DESCRIPTION
         "This object permits management of the table
          by facilitating actions such as row creation,
          construction, and destruction. The value of
          this object has no effect on whether other
          objects in this conceptual row can be
          modified."
   ::= { olsrv2LibLocAttNetSetEntry 5 }


    --
    -- Interface Information Bases  - as defined in
    -- [NHDP], one Interface Information Base for
    -- each OLSRv2 interface.
    --

    -- Note: The IIB is fully defined in the NHDP
    -- specification and its associated MIB.




    --
    -- Neighbor Information Base  - as defined in [NHDP],
    -- extended by the addition of five elements to
    -- each Neighbor Tuple, as defined in Section 8.
    --

    --
    -- Neighbor Set
    --

    olsrv2NibNeighborSetTable OBJECT-TYPE
       SYNTAX       SEQUENCE OF Olsrv2NibNeighborSetEntry
       MAX-ACCESS   not-accessible
```

```
       STATUS       obsolete
       DESCRIPTION
          "A router's Neighbor Set records all network
          addresses of each 1-hop neighbor.  It consists
          of Neighbor Tuples, each representing a single
          1-hop neighbor. "
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2StateGroup 4 }

    olsrv2NibNeighborSetEntry  OBJECT-TYPE
       SYNTAX      Olsrv2NibNeighborSetEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
          "Each Neighbor Tuple in the Neighbor Set, defined
           in [NHDP], has these additional elements:
                N_orig_addr
                N_willingness
                N_mpr
                N_mpr_selector
                N_advertised
           defined here as extensions."
       REFERENCE
          "The OLSRv2 draft."
       INDEX { olsrv2NibNeighborSetRouterId }
    ::= { olsrv2NibNeighborSetTable 1 }

    Olsrv2NibNeighborSetEntry ::=
       SEQUENCE {
          olsrv2NibNeighborSetRouterId
            NeighborRouterId,
          olsrv2NibNeighborSetNIpAddrType
            InetAddressType,
          olsrv2NibNeighborSetNOrigAddr
            InetAddress,
          olsrv2NibNeighborSetNWilliness
            Unsigned32,
          olsrv2NibNeighborSetNMpr
            TruthValue,
          olsrv2NibNeighborSetNMprSelector
            TruthValue,
          olsrv2NibNeighborSetNAdvertised
            TruthValue
         }

    olsrv2NibNeighborSetRouterId  OBJECT-TYPE
       SYNTAX      NeighborRouterId
```

```
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "The object olsrv2NibNeighborSetRouterId is
          the locally assigned ID of the remote router
          referenced in this row.  The IP addrs
          associated with this router is contained
          in the NHDP-MIB's 'nhdpDiscIfSetTable'.
         "
      REFERENCE
         "The OLSRv2 draft."
   ::= { olsrv2NibNeighborSetEntry 1 }

   olsrv2NibNeighborSetNIpAddrType  OBJECT-TYPE
      SYNTAX      InetAddressType
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The type of the olsrv2NibNeighborSetNOrigAddr, as defined
          in the InetAddress MIB [RFC 4001]."
      REFERENCE
         "The OLSRv2 draft."
   ::= { olsrv2NibNeighborSetEntry 2 }

   olsrv2NibNeighborSetNOrigAddr  OBJECT-TYPE
      SYNTAX      InetAddress
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "This is the originator IP address of that
         neighbor."
      REFERENCE
         "The OLSRv2 draft."
   ::= { olsrv2NibNeighborSetEntry 3 }

   olsrv2NibNeighborSetNWilliness  OBJECT-TYPE
      SYNTAX      Unsigned32 (1..7)
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "This object, N_willingness, is the neighbor
          router's willingness to be selected as an MPR, in
          the range from WILL_NEVER (0) to WILL_ALWAYS
          (15), both inclusive."
      REFERENCE
         "The OLSRv2 draft."
   ::= { olsrv2NibNeighborSetEntry 4 }
```

```
   olsrv2NibNeighborSetNMpr  OBJECT-TYPE
      SYNTAX      TruthValue
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "This object, N_mpr, is a boolean flag,
          describing if this neighbor is selected as
          an MPR by this router.

          When set to 'true', this neighbor is selected
          as an MPR by this router.  When set to 'false',
          it is not selected by this router as an MPR."
      REFERENCE
         "The OLSRv2 draft."
   ::= { olsrv2NibNeighborSetEntry 5 }

   olsrv2NibNeighborSetNMprSelector  OBJECT-TYPE
      SYNTAX      TruthValue
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "This object, N_mpr_selector, is a boolean flag,
          describing if this neighbor has selected this router
          as an MPR, i.e. is an MPR selector of this router.

          When set to 'true', then this router is selected as
          an MPR by the neighbor router.  When set to 'false',
          then this router is not selected by the neighbor
          as an MPR"
      REFERENCE
         "The OLSRv2 draft."
   ::= { olsrv2NibNeighborSetEntry 6 }

   olsrv2NibNeighborSetNAdvertised  OBJECT-TYPE
      SYNTAX      TruthValue
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "This object, N_mpr_selector, is a boolean flag, describing if
         this router has elected to advertise a link to this neighbor
         in its TC messages."
      REFERENCE
         "The OLSRv2 draft."
   ::= { olsrv2NibNeighborSetEntry 7 }

   olsrv2NibNeighborSetTableAnsn OBJECT-TYPE
      SYNTAX       Unsigned32
      MAX-ACCESS   read-only
```

```
       STATUS       current
       DESCRIPTION
          "Advertised Neighbor Sequence Number (ANSN), is
          a variable, whose value is included in TC messages to
          indicate the freshness of the information transmitted."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2StateGroup 5 }



    --
    -- Topology Information Base  - this Information
    -- Base is specific to OLSRv2, and is defined in
    -- Section 9.
    --


    --
    -- Advertising Remote Router Set
    --

    olsrv2TibAdRemoteRouterSetTable OBJECT-TYPE
       SYNTAX       SEQUENCE OF Olsrv2TibAdRemoteRouterSetEntry
       MAX-ACCESS   not-accessible
       STATUS       obsolete
       DESCRIPTION
          "A router's Advertising Remote Router Set records
           information describing each remote router in the
           network that transmits TC messages."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2StateGroup 6 }

    olsrv2TibAdRemoteRouterSetEntry  OBJECT-TYPE
       SYNTAX      Olsrv2TibAdRemoteRouterSetEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
          "A router's Advertised Neighbor Set Table entry
           consists of Advertising Remote Router Tuples:

               (AR_orig_addr, AR_seq_number, AR_time)

           Addresses associated with this router are
           found in the NHDP-MIB's 'nhdpDiscIfSetTable'."
       REFERENCE
```

```
          "The OLSRv2 draft."
       INDEX { olsrv2TibAdRemoteRouterSetRouterId }
    ::= { olsrv2TibAdRemoteRouterSetTable 1 }

    Olsrv2TibAdRemoteRouterSetEntry ::=
       SEQUENCE {
          olsrv2TibAdRemoteRouterSetIpAddrType
            InetAddressType,
          olsrv2TibAdRemoteRouterSetIpAddr
            InetAddress,
          olsrv2TibAdRemoteRouterSetRouterId
            NeighborRouterId,
          olsrv2TibAdRemoteRouterSetMaxSeqNo
            Unsigned32,
          olsrv2TibAdRemoteRouterSetExpireTime
            TimeStamp
         }

    olsrv2TibAdRemoteRouterSetIpAddrType  OBJECT-TYPE
       SYNTAX      InetAddressType
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "The type of the olsrv2TibAdRemoteRouterSetIpAddr,
           as defined in the InetAddress MIB [RFC 4001]."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2TibAdRemoteRouterSetEntry 1 }

    olsrv2TibAdRemoteRouterSetIpAddr  OBJECT-TYPE
       SYNTAX      InetAddress
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "This is the originator address of a received
           TC message."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2TibAdRemoteRouterSetEntry 2 }

    olsrv2TibAdRemoteRouterSetRouterId  OBJECT-TYPE
       SYNTAX      NeighborRouterId
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
          "This object is an additional index for each
           Remote Router's IfAddr associated with the
           olsrv2TibAdRemoteRouterSetIpAddr."
```

```
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2TibAdRemoteRouterSetEntry 3 }

     olsrv2TibAdRemoteRouterSetMaxSeqNo  OBJECT-TYPE
        SYNTAX      Unsigned32 (0..65535)
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
           "This is the greatest ANSN in any TC message
            received which originated from the router
            with originator address
            olsrv2TibAdRemoteRouterSetIpAddr."
        REFERENCE
           "The OLSRv2 draft."
    ::= { olsrv2TibAdRemoteRouterSetEntry 4 }

    olsrv2TibAdRemoteRouterSetExpireTime  OBJECT-TYPE
      SYNTAX      TimeStamp
      UNITS       "milliseconds"
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "This is the time at which this
          Tuple expires and MUST be removed."
      REFERENCE
         "The OLSRv2 draft."
   ::= { olsrv2TibAdRemoteRouterSetEntry 5 }



    --
    -- Router Topology Set
    --

    olsrv2TibRouterTopologySetTable OBJECT-TYPE
       SYNTAX       SEQUENCE OF Olsrv2TibTopologySetEntry
       MAX-ACCESS   not-accessible
       STATUS       obsolete
       DESCRIPTION
          "A router's Router Topology Set records topology
           information about the network."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2StateGroup 7 }

    olsrv2TibRouterTopologySetEntry  OBJECT-TYPE
       SYNTAX      Olsrv2TibTopologySetEntry
```

```
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
          "It consists of Router Topology Tuples:

               (TR_from_orig_addr, TR_to_orig_addr,
                    TR_seq_number, TR_time)"
       REFERENCE
          "The OLSRv2 draft."
       INDEX { olsrv2TibRouterTopologySetFromOrigIpAddr }
    ::= { olsrv2TibRouterTopologySetTable 1 }

    Olsrv2TibTopologySetEntry ::=
       SEQUENCE {
          olsrv2TibRouterTopologySetFromOrigIpAddrType
            InetAddressType,
          olsrv2TibRouterTopologySetFromOrigIpAddr
            InetAddress,
          olsrv2TibRouterTopologySetToOrigIpAddrType
            InetAddressType,
          olsrv2TibRouterTopologySetToOrigIpAddr
            InetAddress,
          olsrv2TibRouterTopologySetSeqNo
            Unsigned32,
          olsrv2TibRouterTopologySetExpireTime
            TimeStamp
         }

    olsrv2TibRouterTopologySetFromOrigIpAddrType  OBJECT-TYPE
       SYNTAX      InetAddressType
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "The type of the olsrv2TibRouterTopologySetFromOrigIpAddr,
           as defined in the InetAddress MIB [RFC 4001]."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2TibRouterTopologySetEntry 1 }

    olsrv2TibRouterTopologySetFromOrigIpAddr  OBJECT-TYPE
       SYNTAX      InetAddress
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "This is the originator address of a router which can
          reach the router with originator address TR_to_orig_addr
          in one hop, note that this does not include a prefix length"
       REFERENCE
```

```
          "The OLSRv2 draft."
    ::= { olsrv2TibRouterTopologySetEntry 2 }

    olsrv2TibRouterTopologySetToOrigIpAddrType  OBJECT-TYPE
       SYNTAX      InetAddressType
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "The type of the olsrv2TibRouterTopologySetToOrigIpAddr,
           as defined in the InetAddress MIB [RFC 4001]."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2TibRouterTopologySetEntry 3 }

    olsrv2TibRouterTopologySetToOrigIpAddr  OBJECT-TYPE
       SYNTAX      InetAddress
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "This is the originator address of a router which can be
          reached by the router with originator address
          TR_to_orig_addr in one hop, note that this does
          not include a prefix length."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2TibRouterTopologySetEntry 4 }

     olsrv2TibRouterTopologySetSeqNo  OBJECT-TYPE
       SYNTAX      Unsigned32 (0..65535)
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "This is the greatest ANSN in any TC message
          received which originated from the router
          with originator address TR_from_orig_addr
          (i.e., which contributed to the information
          contained in this Tuple)."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2TibRouterTopologySetEntry 5 }

     olsrv2TibRouterTopologySetExpireTime  OBJECT-TYPE
       SYNTAX      TimeStamp
       UNITS       "milliseconds"
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
          "This is the time at which this
```

```
           Tuple expires and MUST be removed."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2TibRouterTopologySetEntry 6 }



    --
    -- Routable Address Topology Set
    --


    olsrv2TibRoutableAddressTopologySetTable OBJECT-TYPE
       SYNTAX       SEQUENCE OF Olsrv2TibRoutableAddressTopologySetEntry
       MAX-ACCESS   not-accessible
       STATUS       obsolete
       DESCRIPTION
          "A router's Routable Address Topology Set records topology
          information about the routable addresses within the MANET,
          and via which routers they may be reached."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2StateGroup 8 }

    olsrv2TibRoutableAddressTopologySetEntry  OBJECT-TYPE
       SYNTAX      Olsrv2TibRoutableAddressTopologySetEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
          "It consists of Router Topology Tuples:

               (TA_from_orig_addr, TA_to_orig_addr,
                    TA_seq_number, TA_time)"
       REFERENCE
          "The OLSRv2 draft."
       INDEX { olsrv2TibRouterTopologySetFromOrigIpAddr }
    ::= { olsrv2TibRoutableAddressTopologySetTable 1 }

    Olsrv2TibRoutableAddressTopologySetEntry ::=
       SEQUENCE {
          olsrv2TibRoutableAddressTopologySetFromOrigIpAddrType
            InetAddressType,
          olsrv2TibRoutableAddressTopologySetFromOrigIpAddr
            InetAddress,
          olsrv2TibRoutableAddressTopologySetToOrigIpAddrType
            InetAddressType,
          olsrv2TibRoutableAddressTopologySetToOrigIpAddr
            InetAddress,
```

```
          olsrv2TibRoutableAddressTopologySetSeqNo
            Unsigned32,
          olsrv2TibRoutableAddressTopologySetExpireTime
            TimeStamp
         }

    olsrv2TibRoutableAddressTopologySetFromOrigIpAddrType  OBJECT-TYPE
       SYNTAX      InetAddressType
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "The type of the
          olsrv2TibRoutableAddressTopologySetFromOrigIpAddr,
          as defined in the InetAddress MIB [RFC 4001]."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2TibRoutableAddressTopologySetEntry 1 }

   olsrv2TibRoutableAddressTopologySetFromOrigIpAddr  OBJECT-TYPE
       SYNTAX      InetAddress
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "This is the originator address of a router which can
          reach the router with routable address TA_dest_addr
          in one hop."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2TibRoutableAddressTopologySetEntry 2 }

    olsrv2TibRoutableAddressTopologySetToOrigIpAddrType  OBJECT-TYPE
       SYNTAX      InetAddressType
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "The type of the olsrv2TibRouterTopologySetToOrigIpAddr,
           as defined in the InetAddress MIB [RFC 4001]."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2TibRoutableAddressTopologySetEntry 3 }

    olsrv2TibRoutableAddressTopologySetToOrigIpAddr  OBJECT-TYPE
       SYNTAX      InetAddress
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "This is a routable address of a router which can be
          reached by the router with originator address
```

```
          TA_from_orig_addr in one hop."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2TibRoutableAddressTopologySetEntry 4 }

     olsrv2TibRoutableAddressTopologySetSeqNo  OBJECT-TYPE
       SYNTAX      Unsigned32 (0..65535)
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "This is the greatest ANSN in any TC message
          received which originated from the router
          with originator address TA_from_orig_addr
          (i.e., which contributed to the information
          contained in this Tuple)."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2TibRoutableAddressTopologySetEntry 5 }

     olsrv2TibRoutableAddressTopologySetExpireTime  OBJECT-TYPE
       SYNTAX      TimeStamp
       UNITS       "milliseconds"
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "This is the time at which this
           Tuple expires and MUST be removed."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2TibRoutableAddressTopologySetEntry 6 }



    --
    -- Attached Network Set
    --

    olsrv2TibAttNetworksSetTable OBJECT-TYPE
       SYNTAX       SEQUENCE OF Olsrv2TibAttNetworksSetEntry
       MAX-ACCESS   not-accessible
       STATUS       obsolete
       DESCRIPTION
          "A router's Attached Network Set records information
          about networks (which may be outside the MANET)
          attached to other routers and their routable addresses."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2StateGroup 9 }
```

```
    olsrv2TibAttNetworksSetEntry  OBJECT-TYPE
       SYNTAX      Olsrv2TibAttNetworksSetEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "It consists of Attached Network Tuples:

                   (AN_orig_addr, AN_net_addr,
                          AN_dist, AN_seq_number, AN_time)"

       REFERENCE
          "The OLSRv2 draft."
       INDEX { olsrv2TibAttNetworksSetNetIpAddrType,
               olsrv2TibAttNetworksSetNetIpAddr,
               olsrv2TibAttNetworksSetNetIpAddrPrefixLen }
    ::= { olsrv2TibAttNetworksSetTable 1 }

    Olsrv2TibAttNetworksSetEntry ::=
       SEQUENCE {
          olsrv2TibAttNetworksSetOrigIpAddr
            InetAddress,
          olsrv2TibAttNetworksSetNetIpAddrType
            InetAddressType,
          olsrv2TibAttNetworksSetNetIpAddr
            InetAddress,
          olsrv2TibAttNetworksSetNetIpAddrPrefixLen
            InetAddressPrefixLength,
          olsrv2TibAttNetworksSetSeqNo
            Unsigned32,
          olsrv2TibAttNetworksSetDist
            Unsigned32,
          olsrv2TibAttNetworksSetExpireTime
            TimeStamp
         }

   olsrv2TibAttNetworksSetOrigIpAddr  OBJECT-TYPE
      SYNTAX      InetAddress
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "This is the originator address of a
          router which can act as gateway to the
          network with address AN_net_addr,
          note that this does not include a
          prefix length."
      REFERENCE
         "The OLSRv2 draft."
   ::= { olsrv2TibAttNetworksSetEntry 1 }
```

```
    olsrv2TibAttNetworksSetNetIpAddrType  OBJECT-TYPE
       SYNTAX      InetAddressType
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
          "The type of the olsrv2TibAttNetworksSetNetIpAddr,
           as defined in the InetAddress MIB [RFC 4001]."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2TibAttNetworksSetEntry 2 }

    olsrv2TibAttNetworksSetNetIpAddr  OBJECT-TYPE
       SYNTAX      InetAddress
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "This is is the network address of an
           attached network, which may be reached via
           the router with originator address AN_orig_addr."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2TibAttNetworksSetEntry 3 }

    olsrv2TibAttNetworksSetNetIpAddrPrefixLen  OBJECT-TYPE
      SYNTAX      InetAddressPrefixLength
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "Indicates the number of leading one bits that form the
          mask to be logical-ANDed with the destination address
          before being compared to the value in the
          olsrv2TibAttNetworksSetNetIpAddr field."
      REFERENCE
         "The OLSRv2 draft."
   ::= { olsrv2TibAttNetworksSetEntry 4 }

   olsrv2TibAttNetworksSetSeqNo  OBJECT-TYPE
      SYNTAX      Unsigned32 (0..65535)
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The is the greatest ANSN in any TC
          message received which originated from the
          router with originator address AN_orig_addr
          (i.e. which contributed to the information
          contained in this Tuple)."
      REFERENCE
         "The OLSRv2 draft."
```

```
   ::= { olsrv2TibAttNetworksSetEntry 5 }

   olsrv2TibAttNetworksSetDist  OBJECT-TYPE
      SYNTAX      Unsigned32 (0..255)
      UNITS       "hops"
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The is the number of hops to the network
          with address AN_net_addr from the router with
          originator address AN_orig_addr."
      REFERENCE
         "The OLSRv2 draft."
   ::= { olsrv2TibAttNetworksSetEntry 6 }

   olsrv2TibAttNetworksSetExpireTime  OBJECT-TYPE
      SYNTAX      TimeStamp
      UNITS       "milliseconds"
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "This is the time at which this
          Tuple expires and MUST be removed."
      REFERENCE
         "The OLSRv2 draft."
   ::= { olsrv2TibAttNetworksSetEntry 7 }



    --
    -- Routing Set
    --

    olsrv2TibRoutingSetTable OBJECT-TYPE
       SYNTAX       SEQUENCE OF Olsrv2TibRoutingSetEntry
       MAX-ACCESS   not-accessible
       STATUS       obsolete
       DESCRIPTION
          "A router's Routing Set records the first hop along a
          selected path to each destination for which any such
          path is known."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2StateGroup 10 }

    olsrv2TibRoutingSetEntry  OBJECT-TYPE
       SYNTAX      Olsrv2TibRoutingSetEntry
       MAX-ACCESS  not-accessible
```

```
       STATUS      current
       DESCRIPTION
           "It consists of Routing Tuples:

            (R_dest_addr, R_next_iface_addr,
                 R_local_iface_addr, R_dist)"
       REFERENCE
          "The OLSRv2 draft."
       INDEX { olsrv2TibRoutingSetDestIpAddrType,
               olsrv2TibRoutingSetDestIpAddr,
               olsrv2TibRoutingSetDestIpAddrPrefLen }
    ::= { olsrv2TibRoutingSetTable 1 }

    Olsrv2TibRoutingSetEntry ::=
       SEQUENCE {
          olsrv2TibRoutingSetDestIpAddrType
            InetAddressType,
          olsrv2TibRoutingSetDestIpAddr
            InetAddress,
          olsrv2TibRoutingSetDestIpAddrPrefLen
            InetAddressPrefixLength,
          olsrv2TibRoutingSetNextIfIpAddr
            InetAddress,
          olsrv2TibRoutingSetLocalIfIpAddr
            InetAddress,
          olsrv2TibRoutingSetDist
            Unsigned32
         }

    olsrv2TibRoutingSetDestIpAddrType  OBJECT-TYPE
       SYNTAX      InetAddressType
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
          "The type of the olsrv2TibRoutingSetDestIpAddr
           and olsrv2TibRoutingSetNextIfIpAddr,
           as defined in the InetAddress MIB [RFC 4001]."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2TibRoutingSetEntry 1 }

    olsrv2TibRoutingSetDestIpAddr  OBJECT-TYPE
       SYNTAX      InetAddress
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "This is the address of the destination,
           either the address of an interface of
```

```
           a destination router, or the network
           address of an attached network."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2TibRoutingSetEntry 2 }

    olsrv2TibRoutingSetDestIpAddrPrefLen  OBJECT-TYPE
      SYNTAX      InetAddressPrefixLength
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "Indicates the number of leading one bits that form the
          mask to be logical-ANDed with the destination address
          before being compared to the value in the
          olsrv2TibRoutingSetDestNetIpAddr field.

          Note: This definition needs to be consistent
          with the current forwarding table MIB description.
          Specifically, it should allow for longest prefix
          matching of network addresses."
      REFERENCE
         "The OLSRv2 draft."
   ::= { olsrv2TibRoutingSetEntry 3 }

    olsrv2TibRoutingSetNextIfIpAddr  OBJECT-TYPE
       SYNTAX      InetAddress
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "This is the OLSRv2 interface address of the
           'next hop' on the selected path to the
           destination."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2TibRoutingSetEntry 4 }

    olsrv2TibRoutingSetLocalIfIpAddr  OBJECT-TYPE
       SYNTAX      InetAddress
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "This is the address of the local OLSRv2
           interface over which a packet MUST be
           sent to reach the destination by the
           selected path."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2TibRoutingSetEntry 5 }
```

```
    olsrv2TibRoutingSetDist  OBJECT-TYPE
       SYNTAX      Unsigned32 (0..255)
       UNITS       "hops"
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "The is the number of hops on the selected
           path to the destination."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2TibRoutingSetEntry 6 }



    --
    -- Received Message Information Base (RMIB) - records information
    -- required to ensure that a message is processed at most
    -- once and is forwarded at most once per OLSRv2 interface
    -- of a router, using MPR flooding.
    --

    -- Note:  Is it appropriate or necessary to put the
    -- level of detail found in the Processing and
    -- Forwarding Information Base into the OLSRv2-MIB?

    --
    -- Received Set
    --

    olsrv2RmibReceivedSetTable OBJECT-TYPE
       SYNTAX       SEQUENCE OF Olsrv2RmibReceivedSetEntry
       MAX-ACCESS   not-accessible
       STATUS       obsolete
       DESCRIPTION
          "A router has a Received Set per OLSRv2 interface.
          Each Received Set records the signatures of messages
          which have been received over that OLSRv2 interface."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2StateGroup 11 }

    olsrv2RmibReceivedSetEntry  OBJECT-TYPE
       SYNTAX      Olsrv2RmibReceivedSetEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
          "Each consists of Received Tuples:
```

```
          (RX_type, RX_orig_addr, RX_seq_number, RX_time)"
       REFERENCE
          "The OLSRv2 draft."
       INDEX { olsrv2RmibReceivedIfIndex,
               olsrv2RmibReceivedSetOrigIpAddr,
               olsrv2RmibReceivedSetSeqNo }
    ::= { olsrv2RmibReceivedSetTable 1 }

    Olsrv2RmibReceivedSetEntry ::=
       SEQUENCE {
          olsrv2RmibReceivedIfIndex
            InterfaceIndexOrZero,
          olsrv2RmibReceivedSetMsgType
            Unsigned32,
          olsrv2RmibReceivedSetOrigIpAddrType
            InetAddressType,
          olsrv2RmibReceivedSetOrigIpAddr
            InetAddress,
          olsrv2RmibReceivedSetSeqNo
            Unsigned32,
          olsrv2RmibReceivedSetExpireTime
            TimeStamp
         }

    olsrv2RmibReceivedIfIndex  OBJECT-TYPE
       SYNTAX      InterfaceIndexOrZero
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
          "The ID of an interface.  Used for cross
          indexing into other OLSRv2 tables and other
          MIBs."
    ::= { olsrv2RmibReceivedSetEntry 1 }

    olsrv2RmibReceivedSetMsgType  OBJECT-TYPE
       SYNTAX      Unsigned32 (1..255)
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "This is the received Message Type."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2RmibReceivedSetEntry 2 }

    olsrv2RmibReceivedSetOrigIpAddrType  OBJECT-TYPE
       SYNTAX      InetAddressType
       MAX-ACCESS  read-only
       STATUS      current
```

```
       DESCRIPTION
          "The type of the olsrv2RmibReceivedSetOrigIpAddr,
           as defined in the InetAddress MIB [RFC 4001]."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2RmibReceivedSetEntry 3 }

    olsrv2RmibReceivedSetOrigIpAddr  OBJECT-TYPE
       SYNTAX      InetAddress
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "This is the originator address of the received
          message, note that this does not include a
          prefix length."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2RmibReceivedSetEntry 4 }

    olsrv2RmibReceivedSetSeqNo  OBJECT-TYPE
       SYNTAX      Unsigned32 (0..65535)
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "This is the message sequence number of the received
          message."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2RmibReceivedSetEntry 5 }

    olsrv2RmibReceivedSetExpireTime  OBJECT-TYPE
       SYNTAX      TimeStamp
       UNITS       "milliseconds"
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "This specifies the time at which this Tuple
          expires and MUST be removed."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2RmibReceivedSetEntry 6 }



    --
    -- Processed Set
    --
```

```
    olsrv2RmibProcessedSetTable OBJECT-TYPE
       SYNTAX       SEQUENCE OF Olsrv2RmibProcessedSetEntry
       MAX-ACCESS   not-accessible
       STATUS       obsolete
       DESCRIPTION
          "A router has a single Processed Set which
          records signatures of messages which have
          been processed by the router."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2StateGroup 12 }

    olsrv2RmibProcessedSetEntry  OBJECT-TYPE
       SYNTAX      Olsrv2RmibProcessedSetEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
          "Each consists of Processed Tuples:

          (P_type, P_orig_addr, P_seq_number, P_time)"
       REFERENCE
          "The OLSRv2 draft."
       INDEX { olsrv2RmibProcessedSetOrigIpAddr,
               olsrv2RmibProcessedSetSeqNo }
    ::= { olsrv2RmibProcessedSetTable 1 }

    Olsrv2RmibProcessedSetEntry ::=
       SEQUENCE {
          olsrv2RmibProcessedSetMsgType
            Unsigned32,
          olsrv2RmibProcessedSetOrigIpAddrType
            InetAddressType,
          olsrv2RmibProcessedSetOrigIpAddr
            InetAddress,
          olsrv2RmibProcessedSetSeqNo
            Unsigned32,
          olsrv2RmibProcessedSetExpireTime
            TimeStamp
         }

    olsrv2RmibProcessedSetMsgType  OBJECT-TYPE
       SYNTAX      Unsigned32 (1..255)
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "This is the processed Message Type."
       REFERENCE
          "The OLSRv2 draft."
```

```
    ::= { olsrv2RmibProcessedSetEntry 1 }

    olsrv2RmibProcessedSetOrigIpAddrType  OBJECT-TYPE
       SYNTAX      InetAddressType
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "The type of the olsrv2RmibProcessedSetOrigIpAddr, as defined
           in the InetAddress MIB [RFC 4001]."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2RmibProcessedSetEntry 2 }

    olsrv2RmibProcessedSetOrigIpAddr  OBJECT-TYPE
       SYNTAX      InetAddress
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "This is the originator address of the processed
          message, note that this does not include a
          prefix length."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2RmibProcessedSetEntry 3 }

    olsrv2RmibProcessedSetSeqNo  OBJECT-TYPE
       SYNTAX      Unsigned32 (0..65535)
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "This is the message sequence number of the processed
          message."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2RmibProcessedSetEntry 4 }

    olsrv2RmibProcessedSetExpireTime  OBJECT-TYPE
       SYNTAX      TimeStamp
       UNITS       "milliseconds"
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "This specifies the time at which this Tuple
          expires and MUST be removed."
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2RmibProcessedSetEntry 5 }
```

```
--
-- Forwarded Set
--

olsrv2RmibForwardedSetTable OBJECT-TYPE
   SYNTAX       SEQUENCE OF Olsrv2RmibForwardedSetEntry
   MAX-ACCESS   not-accessible
   STATUS       obsolete
   DESCRIPTION
      "A router has a single Forwarded Set which records
      signatures of messages which have been forwarded by
      the router."
   REFERENCE
      "The OLSRv2 draft."
::= { olsrv2StateGroup 13 }

olsrv2RmibForwardedSetEntry  OBJECT-TYPE
   SYNTAX      Olsrv2RmibForwardedSetEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
      "Each consists of Forwarded Tuples:

       (F_type, F_orig_addr, F_seq_number, F_time)"
   REFERENCE
      "The OLSRv2 draft."
   INDEX { olsrv2RmibReceivedSetOrigIpAddr,
           olsrv2RmibReceivedSetSeqNo }
::= { olsrv2RmibForwardedSetTable 1 }

Olsrv2RmibForwardedSetEntry ::=
   SEQUENCE {
      olsrv2RmibForwardedSetMsgType
        Unsigned32,
      olsrv2RmibForwardedSetOrigIpAddrType
        InetAddressType,
      olsrv2RmibForwardedSetOrigIpAddr
        InetAddress,
      olsrv2RmibForwardedSetSeqNo
        Unsigned32,
      olsrv2RmibForwardedSetExpireTime
        TimeStamp
     }

olsrv2RmibForwardedSetMsgType  OBJECT-TYPE
   SYNTAX      Unsigned32 (1..255)
   MAX-ACCESS  read-only
   STATUS      current
```

```
      DESCRIPTION
         "This is the forwarded Message Type."
      REFERENCE
         "The OLSRv2 draft."
   ::= { olsrv2RmibForwardedSetEntry 1 }

   olsrv2RmibForwardedSetOrigIpAddrType  OBJECT-TYPE
      SYNTAX      InetAddressType
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The type of the olsrv2RmibForwardedSetOrigIpAddr,
          as defined in the InetAddress MIB [RFC 4001]."
      REFERENCE
         "The OLSRv2 draft."
   ::= { olsrv2RmibForwardedSetEntry 2 }

   olsrv2RmibForwardedSetOrigIpAddr  OBJECT-TYPE
      SYNTAX      InetAddress
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "This is the originator address of the forwarded
         message, note that this does not include a
         prefix length."
      REFERENCE
         "The OLSRv2 draft."
   ::= { olsrv2RmibForwardedSetEntry 3 }

   olsrv2RmibForwardedSetSeqNo  OBJECT-TYPE
      SYNTAX      Unsigned32 (0..65535)
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "This is the message sequence number of the forwarded
         message."
      REFERENCE
         "The OLSRv2 draft."
   ::= { olsrv2RmibForwardedSetEntry 4 }

   olsrv2RmibForwardedSetExpireTime  OBJECT-TYPE
      SYNTAX      TimeStamp
      UNITS       "milliseconds"
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "This specifies the time at which this Tuple
         expires and MUST be removed."
```

```
       REFERENCE
          "The OLSRv2 draft."
    ::= { olsrv2RmibForwardedSetEntry 5 }



   --
   -- OLSRv2 Performance Group
   --
   --    Contains objects which help to characterize the
   --    performance of the OLSRv2 routing process.
   --

   olsrv2PerformanceObjGrp  OBJECT IDENTIFIER ::= { olsrv2MIBObjects 3 }


  --
  -- Objects per local interface
  --

  olsrv2InterfacePerfTable  OBJECT-TYPE
      SYNTAX      SEQUENCE OF Olsrv2InterfacePerfEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "This table summarizes performance objects that are
          measured per local OLSRv2 interface."
      REFERENCE
         "The OLSRv2 draft."
   ::= { olsrv2PerformanceObjGrp 1 }

   olsrv2InterfacePerfEntry OBJECT-TYPE
      SYNTAX      Olsrv2InterfacePerfEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "A single entry contains performance counters for
          a local OLSRv2 interface."
      INDEX { olsrv2IfPerfIndex }
   ::= { olsrv2InterfacePerfTable 1 }

   Olsrv2InterfacePerfEntry ::=
      SEQUENCE {
         olsrv2IfPerfIndex
            InterfaceIndexOrZero,
         olsrv2IfTcMessageXmits
            Counter32,
         olsrv2IfTcMessageRecvd
```

```
            Counter32,
         olsrv2IfTcMessageXmitAccumulatedSize
            Counter32,
         olsrv2IfTcMessageRecvdAccumulatedSize
            Counter32,
         olsrv2IfTcMessageTriggeredXmits
            Counter32,
         olsrv2IfTcMessagePeriodicXmits
            Counter32,
         olsrv2IfTcMessageForwardedXmits
            Counter32,
         olsrv2IfTcMessageXmitAccumulatedMPRSelectorCount
            Counter32
         }

   olsrv2IfPerfIndex  OBJECT-TYPE
      SYNTAX      InterfaceIndexOrZero
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "The ID of an interface.  Used for cross
          indexing into other OLSRv2 tables and other
          MIBs."
   ::= { olsrv2InterfacePerfEntry 1 }

   olsrv2IfTcMessageXmits  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter is incremented each time a TC
         message has been transmitted on that interface."
   ::= { olsrv2InterfacePerfEntry 2 }

   olsrv2IfTcMessageRecvd  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter is incremented each time a
         TC message has been received on that interface."
   ::= { olsrv2InterfacePerfEntry 3 }

   olsrv2IfTcMessageXmitAccumulatedSize  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
```

```
         "A counter is incremented by the number of octets in
         a TC message each time a
         TC message has been sent."
   ::= { olsrv2InterfacePerfEntry 4 }

   olsrv2IfTcMessageRecvdAccumulatedSize  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter is incremented by the number of octets in
         a TC message each time a
         TC message has been received."
   ::= { olsrv2InterfacePerfEntry 5 }

   olsrv2IfTcMessageTriggeredXmits  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter is incremented each time a triggered
         TC message has been sent."
   ::= { olsrv2InterfacePerfEntry 6 }

   olsrv2IfTcMessagePeriodicXmits  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter is incremented each time a periodic
         TC message has been sent."
   ::= { olsrv2InterfacePerfEntry 7 }

   olsrv2IfTcMessageForwardedXmits  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter is incremented each time a
         TC message has been forwarded."
   ::= { olsrv2InterfacePerfEntry 8 }

   olsrv2IfTcMessageXmitAccumulatedMPRSelectorCount OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter is incremented by the number of advertised
```

```
         MPR selectors in a TC each time a TC
         message has been sent."
   ::= { olsrv2InterfacePerfEntry 9 }



   --
   -- Objects concerning the Routing set
   --

   olsrv2RoutingSetRecalculationCount  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "This counter increments each time the Routing Set has
         been recalculated."
   ::= { olsrv2PerformanceObjGrp 2 }

   --
   -- Objects concerning the MPR set
   --

   olsrv2MPRSetRecalculationCount  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "This counter increments each time the MPRs
         of this router have been recalculated."
   ::= { olsrv2PerformanceObjGrp 3 }




   --
   -- Notifications
   --

olsrv2NotificationsControl OBJECT IDENTIFIER ::=
                                     { olsrv2MIBNotifications 1 }
olsrv2NotificationsObjects OBJECT IDENTIFIER ::=
                                     { olsrv2MIBNotifications 2 }
olsrv2NotificationsStates  OBJECT IDENTIFIER ::=
                                     { olsrv2MIBNotifications 3 }
```

```
   -- olsrv2NotificationsControl

   olsrv2SetNotification OBJECT-TYPE
          SYNTAX       OCTET STRING (SIZE(4))
          MAX-ACCESS   read-write
          STATUS       current
          DESCRIPTION
             "A 4-octet string serving as a bit map for
             the notification events defined by the OLSRv2
             notifications. This object is used to enable
             and disable specific OLSRv2 notifications where
             a 1 in the bit field represents enabled. The
             right-most bit (least significant) represents
             notification 1.

             This object is persistent and when written
             the entity SHOULD save the change to
             non-volatile storage.
             "
           ::= { olsrv2NotificationsControl 1 }

   olsrv2RoutingSetRecalculationCountThreshold OBJECT-TYPE
          SYNTAX       Integer32 (0..255)
          MAX-ACCESS   read-write
          STATUS       current
          DESCRIPTION
             "A threshold value for the
              olsrv2RoutingSetRecalculationCount object.
              If the number of occurrences exceeds this
              threshold within the previous
              olsrv2RoutingSetReculculationCountWindow,
              then the olsrv2RoutingSetRecalculationCountChange
              notification is to be sent.
             "
           ::= { olsrv2NotificationsControl 2 }

   olsrv2RoutingSetRecalculationCountWindow OBJECT-TYPE
          SYNTAX       TimeTicks
          MAX-ACCESS   read-write
          STATUS       current
          DESCRIPTION
             "A time window for the
              olsrv2RoutingSetRecalculationCount object.
              If the number of occurrences exceeds the
              olsrv2RoutingSetRecalculationCountThreshold
              within the previous
              olsrv2RoutingSetRecalculationCountWindow,
              then the
```

```
              olsrv2RoutingSetRecalculationCountChange
              notification is to be sent.

              This object represents the time in hundredths
              of a second.
             "
           ::= { olsrv2NotificationsControl 3 }

   olsrv2MPRSetRecalculationCountThreshold OBJECT-TYPE
          SYNTAX       Integer32 (0..255)
          MAX-ACCESS   read-write
          STATUS       current
          DESCRIPTION
             "A threshold value for the
              olsrv2MPRSetRecalculationCount object.
              If the number of occurrences exceeds this
              threshold within the previous
              olsrv2MPRSetReculculationCountWindow,
              then the
              olsrv2MPRSetRecalculationCountChange
              notification is to be sent.
             "
           ::= { olsrv2NotificationsControl 4 }

   olsrv2MPRSetRecalculationCountWindow OBJECT-TYPE
          SYNTAX       TimeTicks
          MAX-ACCESS   read-write
          STATUS       current
          DESCRIPTION
             "A time window for the
              olsrv2MPRSetRecalculationCount object.
              If the number of occurrences exceeds the
              olsrv2MPRSetRecalculationCountThreshold
              within the previous
              olsrv2MPRSetRecalculationCountWindow,
              then the
              olsrv2MPRSetRecalculationCountChange
              notification is to be sent.

              This object represents the time in hundredths
              of a second.
             "
           ::= { olsrv2NotificationsControl 5 }


   -- olsrv2NotificationsObjects

   olsrv2RouterStatusChange NOTIFICATION-TYPE
```

```
          OBJECTS { olsrv2OrigIpAddrType, -- The address type of
                                          -- the originator of
                                          --   the notification.
                    olsrv2OrigIpAddr,     -- The originator of
                                          --   the notification.
                    olsrv2RouterStatus    -- The new state.
                  }
          STATUS       current
          DESCRIPTION
             "olsrv2RouterStatusChange is a notification sent
              when a the OLSRv2 router changes it status.
              The router status is maintained in the
              olsrv2RouterStatus object.
             "
          ::= { olsrv2NotificationsObjects 1 }

   olsrv2OrigIpAddrChange NOTIFICATION-TYPE
          OBJECTS { olsrv2OrigIpAddrType, -- The address type of
                                          -- the originator of
                                          --   the notification.
                    olsrv2OrigIpAddr,     -- The originator of
                                          --    the notification.
                    olsrv2PreviousOrigIpAddrType, -- The address
                                          -- type of previous
                                          -- address of
                                          -- the originator of
                                          --   the notification.
                    olsrv2PreviousOrigIpAddr  -- The previous
                                          -- address of the
                                          -- originator of
                                          --    the notification.
                  }
          STATUS       current
          DESCRIPTION
             "olsrv2RouterStatusChange is a notification sent when a
              the OLSRv2 router changes it status.  The router
              status is maintained in the olsrv2RouterStatus
              object.
             "
          ::= { olsrv2NotificationsObjects 2 }

    olsrv2RoutingSetRecalculationCountChange NOTIFICATION-TYPE
          OBJECTS { olsrv2OrigIpAddrType, -- The address type of
                                          -- the originator of
                                          --   the notification.
                    olsrv2OrigIpAddr,     -- The originator of
                                          --   the notification.
                    olsrv2RoutingSetRecalculationCount  -- The
```

```
                                      -- new count of the
                                      -- routing set
                                      -- recalculations.
              }
           STATUS       current
           DESCRIPTION
              "olsrv2RoutingSetRecalculationCountChange is
               a notification sent when a significant number of
               routing set recalculations have occurred.
               The network administrator should select
               appropriate values for 'significant number of
               neighbors' and 'short time' through the settings
               of the olsrv2RoutingSetRecalculationCountThreshold
               and olsrv2RoutingSetRecalculationCountWindow
               objects.
              "
           ::= { olsrv2NotificationsObjects 3 }

   olsrv2MPRSetRecalculationCountChange NOTIFICATION-TYPE
           OBJECTS { olsrv2OrigIpAddrType, -- The address type of
                                      --   the originator of
                                      --   the notification.
                     olsrv2OrigIpAddr,     -- The originator of
                                      --   the notification.
                     olsrv2MPRSetRecalculationCount  -- The new
                                      --   MPR set
                                      --   recalculation
                                      --   count.
                    }
           STATUS       current
           DESCRIPTION
              "olsrv2MPRSetRecalculationCountChange is
               a notification sent when a significant number of
               MPR set recalculations have occurred.
               The network administrator should select
               appropriate values for 'significant number of
               neighbors' and 'short time' through the settings
               of the olsrv2MPRSetRecalculationCountThreshold
               and olsrv2MPRSetRecalculationCountWindow
               objects.
              "
           ::= { olsrv2NotificationsObjects 4 }


    -- olsrv2NotificationStates

    olsrv2PreviousOrigIpAddrType  OBJECT-TYPE
```

```
      SYNTAX      InetAddressType
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The type of the olsrv2PreviousOrigIpAddr,
          as defined in the InetAddress MIB [RFC 4001].

          This objected should be updated each time the
          olsrv2OrigIpAddrType is changed.

          This object is persistent and when written
          the entity SHOULD save the change to
          non-volatile storage.
         "
      REFERENCE
         "The OLSRv2 draft."
   ::= { olsrv2NotificationsStates 1 }

   olsrv2PreviousOrigIpAddr  OBJECT-TYPE
      SYNTAX      InetAddress
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The previous origination IP address
          of this OLSRv2 router.

          This object should be updated each time
          the olsrv2OrigIpAddr is modified.

          This object is persistent and when written
          the entity SHOULD save the change to
          non-volatile storage.
      "
      REFERENCE
         "The OLSRv2 draft."
   ::= { olsrv2NotificationsStates 2 }



  --
  -- Compliance Statements
  --


  olsrv2Compliances  OBJECT IDENTIFIER ::= { olsrv2MIBConformance 1 }
  olsrv2MIBGroups    OBJECT IDENTIFIER ::= { olsrv2MIBConformance 2 }

  olsrv2BasicCompliance  MODULE-COMPLIANCE
```

```
      STATUS current
      DESCRIPTION "The basic implementation requirements for
                   managed network entities that implement
                   the OLSRv2 routing process."
      MODULE  -- this module
      MANDATORY-GROUPS { olsrv2ConfigObjectsGroup }
   ::= { olsrv2Compliances 1 }

   olsrv2FullCompliance MODULE-COMPLIANCE
      STATUS current
      DESCRIPTION "The full implementation requirements for
                   managed network entities that implement
                   the OLSRv2 routing process."
      MODULE  -- this module
      MANDATORY-GROUPS { olsrv2ConfigObjectsGroup,
                         olsrv2StateObjectsGroup,
                         olsrv2PerfObjectsGroup,
                         olsrv2NotificationsObjectsGroup,
                         olsrv2NotificationsGroup }
   ::= { olsrv2Compliances 2 }

   --
   -- Units of Conformance
   --

   olsrv2ConfigObjectsGroup OBJECT-GROUP
      OBJECTS {
              olsrv2OrigIpAddrType,
              olsrv2OrigIpAddr,
              olsrv2OHoldTime,
              olsrv2TcInterval,
              olsrv2TcMinInterval,
              olsrv2THoldTime,
              olsrv2AHoldTime,
              olsrv2RxHoldTime,
              olsrv2PHoldTime,
              olsrv2FHoldTime,
              olsrv2TpMaxJitter,
              olsrv2TtMaxJitter,
              olsrv2FMaxJitter,
              olsrv2TcHopLimit,
              olsrv2Willingness
      }
      STATUS  current
      DESCRIPTION
         "Set of OLSRv2 configuration objects implemented
          in this module."
   ::= { olsrv2MIBGroups 1 }
```

```
   olsrv2StateObjectsGroup  OBJECT-GROUP
      OBJECTS {
              olsrv2RouterStatus,
              olsrv2LibOrigSetIpAddrType,
              olsrv2LibOrigSetIpAddr,
              olsrv2LibLocAttNetSetIpAddrType,
              olsrv2LibLocAttNetSetIpAddr,
              olsrv2LibLocAttNetSetIpAddrPrefixLen,
              olsrv2LibLocAttNetSetDistance,
              olsrv2LibLocAttNetSetRowStatus,
              olsrv2NibNeighborSetNIpAddrType,
              olsrv2NibNeighborSetNOrigAddr,
              olsrv2NibNeighborSetNWilliness,
              olsrv2NibNeighborSetNMpr,
              olsrv2NibNeighborSetNMprSelector,
              olsrv2NibNeighborSetNAdvertised,
              olsrv2NibNeighborSetTableAnsn,
              olsrv2TibAdRemoteRouterSetIpAddrType,
              olsrv2TibAdRemoteRouterSetIpAddr,
              olsrv2TibAdRemoteRouterSetMaxSeqNo,
              olsrv2TibRouterTopologySetFromOrigIpAddrType,
              olsrv2TibRouterTopologySetFromOrigIpAddr,
              olsrv2TibRouterTopologySetToOrigIpAddrType,
              olsrv2TibRouterTopologySetToOrigIpAddr,
              olsrv2TibRouterTopologySetSeqNo,
              olsrv2TibRoutableAddressTopologySetExpireTime,
              olsrv2TibRoutableAddressTopologySetFromOrigIpAddrType,
              olsrv2TibRoutableAddressTopologySetFromOrigIpAddr,
              olsrv2TibRoutableAddressTopologySetToOrigIpAddrType,
              olsrv2TibRoutableAddressTopologySetToOrigIpAddr,
              olsrv2TibRoutableAddressTopologySetSeqNo,
              olsrv2TibAttNetworksSetOrigIpAddr,
              olsrv2TibAttNetworksSetNetIpAddr,
              olsrv2TibAttNetworksSetNetIpAddrPrefixLen,
              olsrv2TibAttNetworksSetSeqNo,
              olsrv2TibAttNetworksSetDist,
              olsrv2TibAttNetworksSetExpireTime,
              olsrv2TibRoutingSetDestIpAddr,
              olsrv2TibRoutingSetDestIpAddrPrefLen,
              olsrv2TibRoutingSetNextIfIpAddr,
              olsrv2TibRoutingSetLocalIfIpAddr,
              olsrv2TibRoutingSetDist,
              olsrv2RmibReceivedSetMsgType,
              olsrv2RmibReceivedSetOrigIpAddrType,
              olsrv2RmibReceivedSetOrigIpAddr,
              olsrv2RmibReceivedSetSeqNo,
              olsrv2RmibReceivedSetExpireTime,
              olsrv2RmibProcessedSetMsgType,
```

```
              olsrv2RmibProcessedSetOrigIpAddrType,
              olsrv2RmibProcessedSetOrigIpAddr,
              olsrv2RmibProcessedSetSeqNo,
              olsrv2RmibProcessedSetExpireTime,
              olsrv2RmibForwardedSetMsgType,
              olsrv2RmibForwardedSetOrigIpAddrType,
              olsrv2RmibForwardedSetOrigIpAddr,
              olsrv2RmibForwardedSetSeqNo,
              olsrv2RmibForwardedSetExpireTime
      }
      STATUS  current
      DESCRIPTION
         "Set of OLSRv2 state objects implemented
          in this module."
   ::= { olsrv2MIBGroups 2 }

   olsrv2PerfObjectsGroup  OBJECT-GROUP
      OBJECTS {
              olsrv2IfTcMessageXmits,
              olsrv2IfTcMessageRecvd,
              olsrv2IfTcMessageXmitAccumulatedSize,
              olsrv2IfTcMessageRecvdAccumulatedSize,
              olsrv2IfTcMessageTriggeredXmits,
              olsrv2IfTcMessagePeriodicXmits,
              olsrv2IfTcMessageForwardedXmits,
              olsrv2IfTcMessageXmitAccumulatedMPRSelectorCount,
              olsrv2RoutingSetRecalculationCount,
              olsrv2MPRSetRecalculationCount
      }
      STATUS  current
      DESCRIPTION
         "Set of OLSRv2 performance objects implemented
          in this module by total and per interface."
   ::= { olsrv2MIBGroups 3 }

    olsrv2NotificationsObjectsGroup OBJECT-GROUP
      OBJECTS {
            olsrv2SetNotification,
            olsrv2RoutingSetRecalculationCountThreshold,
            olsrv2RoutingSetRecalculationCountWindow,
            olsrv2MPRSetRecalculationCountThreshold,
            olsrv2MPRSetRecalculationCountWindow,
            olsrv2PreviousOrigIpAddrType,
            olsrv2PreviousOrigIpAddr
      }
      STATUS current
      DESCRIPTION
      "Set of OLSRv2 notification objects implemented
```

```
      in this module."
   ::= { olsrv2MIBGroups 4 }


    olsrv2NotificationsGroup OBJECT-GROUP
      OBJECTS {
            olsrv2RouterStatusChange,
            olsrv2OrigIpAddrChange,
            olsrv2RoutingSetRecalculationCountChange,
            olsrv2MPRSetRecalculationCountChange
      }
      STATUS current
      DESCRIPTION
      "Set of OLSRv2 notifications implemented
      in this module."
   ::= { olsrv2MIBGroups 5 }



   END
```

8. Security Considerations

This MIB defines objects for the configuration, monitoring and notification of the Optimized Link State Routing protocol version 2 [OLSRv2]. OLSRv2 allows routers to acquire topological information of the routing domain by virtue of exchanging TC message, to calculate shortest paths to each destination router in the routing domain, to select relays for network-wide transmissions etc.

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 can have a negative effect on network operations. These are the tables and objects and their sensitivity/vulnerability:

- olsrv2TcInterval, olsrv2TcMinInterval - these writable objects control the rate at which TC messages are sent. If set at too high a rate, this could represent a form of DOS attack by overloading interface resources. If set low, OLSRv2 may not converge fast enough to provide accurate routes to all destinations in the routing domain.

- olsrv2TcHopLimit - defines the hop limit for TC messages. If set too low, messages will not be forwarded beyond the defined scope, and thus routers further away from the message originator will not be able to construct appropriate topology graphs.

- o olsrv2OHoldTime, olsrv2THoldTime, olsrv2AHoldTime, olsrv2RxHoldTime, olsrv2PHoldTime, olsrv2FHoldTime - define hold times for tuples of different Information Bases of OLSRv2. If set too low, information will expire quickly, and may this harm a correct operation of the routing protocol.

- o olsrv2Willingness - defines the willingness of this router to become MPR. If this is set to WILL_NEVER (0), the managed router will not forward any TC messages, nor accept a selection to become MPR by neighboring routers. If set to WILL_ALWAYS (15), the router will be preferred by neighbors during MPR selection, and may thus attract more traffic.

- o olsrv2TpMaxJitter, olsrv2TtMaxJitter, olsrv2FMaxJitter - define jitter values for TC message transmission and forwarding. If set too low, control traffic may get lost if the channel is lossy.

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:

- o olsrv2TibRouterTopologySetTable - The contains information on the topology of the MANET, specifically the IP address of the routers in the MANET (as identified by olsrv2TibRouterTopologySetFromOrigIpAddr and olsrv2TibRouterTopologySetToOrigIpAddr objects). This information provides an adversary broad information on the members of the MANET, located within this single table. This information can be use to expedite attacks on the other members of the MANET without having to go through a laborious discovery process on their own. olsrv2TibRouterTopologySetFromOrigIpAddr is the index into the table, and has a MAX-ACCESS of 'not-accessible'. However, this information can be exposed using SNMP operations.

MANET technology is often deployed to support communications of emergency services or military tactical applications. In these applications, it is imperative to maintain the proper operation of the communications network and to protect sensitive information related to its operation. Therefore, when implementing these capabilities, the full use of SNMPv3 cryptographic mechanisms for authentication and privacy is RECOMMENDED.

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.

It is RECOMMENDED that implementers consider the security features as provided by the SNMPv3 framework (see [RFC3410], Section 8, including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

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.

9. IANA Considerations

This memo does not include any request to IANA.

10. References

10.1. Normative References

[RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.

[RFC3418] Presuhn, R., "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3418, December 2002.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

[OLSRv2] Clausen, T., Dearlove, C., and P. Jacquet, "The Optimized Link State Routing Protocol version 2", draft-ietf-manet-olsr-11 (work in progress), April 2010.

[NHDP] Clausen, T., Dearlove, C., and J. Dean, "Mobile Ad Hoc Network (MANET) Neighborhood Discovery Protocol (NHDP)", draft-ietf-manet-nhdp-13 (work in progress), July 2010.

[NHDP-MIB] Herberg, U., Cole, R., and I. Chakeres, "Definition of Managed Objects for the Neighborhood Discovery Protocol", draft-ietf-manet-nhdp-mib-04 (work in progress), July 2010.

[RFC4001] Daniele, M., Haberman, B., Routhier, S., and J. Schoenwaelder, "Textual Conventions for Internet Network Addresses", RFC 4001, February 2005.

[RFC3781] Strauss, F. and J. Schoenwaelder, "Next Generation Structure of Management Information (SMIng) Mappings to the Simple Network Management Protocol (SNMP)", RFC 3781, May 2004.

10.2. Informative References

[RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

[REPORT] Cole, R., Macker, J., and A. Morton, "Definition of Managed Objects for Performance Reporting", draft-ietf-manet-report-mib-00 (work in progress), July 2010.

Appendix A. Change Log

This section identifies the changes made during the development of this MIB.

Here we list the changes made in developing draft-ietf-manet-olsrv2-mib-03.

1. Added the NotificationGroup and updated Conformance to reflect these additions.

2. Cleaned up some of the text associated with 'Derived Objects' within the Performance Group discussion within the introductory text.

3. Added the olsrv2OrigIpAddrType and olsrv2OrigIpAddr objects to the Configuration Group to configure and hold the router ID.

Here we list the changes made in developing draft-ietf-manet-olsrv2-mib-02.

1. Shortened text about the Configuration Group and the State Group.

2. Made coherent with NHDP-MIB.

3. Cleaned up errors.

4. Added Security Considerations section.

5. Updated "Relations to other MIBs" section.

6. Added Notifications section (but no notifications defined yet).

7. Changed type of several objects in the MIB (for timers).

8. Added information identifying objects requiring non-volatile storage within the DESCRIPTION clause of the objects within the OLSRv2-MIB.

Here we list the changes made in developing draft-ietf-manet-olsrv2-mib-01.

1. Added Performance Group objects

2. Updated draft to adhere to the current version of the OLSRv2 draft.

3. Cleaned up errors.

4. Added U. Herberg as new author.

Here we list the changes made in developing draft-ietf-manet-olsrv2-mib-00.

1. Rev'd the draft as a new working group document.

2. Ran 'smilint' against the module and cleaned up syntax errors and other issues discovered by the checker.

Here we list the changes made in developing draft-cole-manet-olsr-mib-01.

1. Completely reworked the entire Configuration Objects group in order to align with the newly developed NHDP-MIB draft.

Appendix B. Open Issues

This section contains the set of open issues related to the development and design of the OLSRv2-MIB. This section will not be present in the final version of the MIB and will be removed once all the open issues have been resolved.

1. Specify specific SNMP response to the snmp set request, i.e., 'generic error', 'bad value', etc.

2. Run through the MIB checker.

Appendix C. Note to the RFC Editor

```
**************************************************************
* Note to the RFC Editor (to be removed prior to publication) *
*                                                            *
* 1) The reference to RFCXXXX within the DESCRIPTION clauses  *
* of the MIB module point to this draft and are to be         *
* assigned by the RFC Editor.                                 *
*                                                            *
* 2) The reference to RFCXXX2 throughout this document point  *
* to the current draft-ietf-manet-olsrv2-xx.txt.  This        *
* need to be replaced with the XXX RFC number.                *
*                                                            *
**************************************************************
```

Authors' Addresses

Ulrich Herberg
LIX, Ecole Polytechnique
Palaiseau Cedex, 91128
France

EMail: ulrich@herberg.name
URI: http://www.herberg.name/

Robert G. Cole
US Army CERDEC
328 Hopkins Road, Bldg 245
Aberdeen Proving Ground, Maryland 21005
USA

Phone: +1 410 278 6779
EMail: robert.g.cole@us.army.mil
URI: http://www.cs.jhu.edu/~rgcole/

Thomas Heide Clausen
LIX, Ecole Polytechnique
Palaiseau Cedex, 91128
France

Phone: +33 6 6058 9349
EMail: T.Clausen@computer.org
URI: http://www.ThomasClausen.org/

MANET Cryptographical Signature TLV Definition
draft-ietf-manet-packetbb-sec-03

Abstract

This document describes general and flexible TLVs (type-length-value structure) for representing cryptographic signatures as well as timestamps, using the generalized MANET packet/message format [RFC5444]. It defines two Packet TLVs, two Message TLVs, and two Address Block TLVs, for affixing cryptographic signatures and timestamps to a packet, message and address, respectively.

Table of Contents

1. Introduction

This document specifies:

- o two TLVs for carrying cryptographic signatures and timestamps in packets, messages and address blocks as defined by [RFC5444],
- o how cryptographic signatures are calculated, taking (for Message TLVs) into account the mutable message header fields (<msg-hop-limit> and <msg-hop-count>) where these fields are present in messages.

This document requests from IANA:

- o allocations for these Packet, Message, and Address Block TLVs from the 0-223 Packet TLV range, the 0-127 Message TLV range and the 0-127 Address Block TLV range from [RFC5444],
- o creation of two IANA registries for recording code points for hash function and signature calculation, respectively.

2. Terminology

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 [RFC2119].

This document uses the terminology and notation defined in [RFC5444].

3. Applicability Statement

MANET routing protocols using the format defined in [RFC5444] are accorded the ability to carry additional information in control messages and packets, through inclusion of TLVs. Information so included MAY be used by a routing protocol, or by an extension of a routing protocol, according to its specification.

This document specifies how to include a cryptographic signature for a packet, message or address by way of such TLVs. This document also specifies how to treat "mutable" fields (<msg-hop-count> and <msg-hop-limit>), if present, in the message header when calculating signatures, such that the resulting signature can be correctly verified by any recipient, and how to include this signature.

4. Security Architecture

Basic MANET routing protocol specifications are often "oblivious to security", however have a clause allowing a control message to be rejected as "badly formed" prior to it being processed or forwarded. Protocols such as [NHDP] and [OLSRv2] recognize external reasons (such as failure to verify a signature) for rejecting a message as "badly formed", and therefore "invalid for processing". This architecture is a result of the observation that with respect to security in MANETs, "one size rarely fits all" and that MANET routing protocol deployment domains have varying security requirements ranging from "unbreakable" to "virtually none". The virtue of this approach is that MANET routing protocol specifications (and implementations) can remain "generic", with extensions providing proper deployment-domain specific security mechanisms.

The MANET routing protocol "security architecture", in which this specification situates itself, can therefore be summarized as follows:

- Security-oblivious MANET routing protocol specifications, with a clause allowing an extension to reject a message (prior to processing/forwarding) as "badly formed".

- MANET routing protocol security extensions, rejecting messages as "badly formed", as appropriate for a given deployment-domain specific security requirement.

- Code-points and an exchange format for information, necessary for specification of such MANET routing protocol security extensions.

This document addresses the last of these issues, by specifying a common exchange format for cryptographic signatures, making reservations from within the Packet TLV, Message TLV and Address Block TLV registries of [RFC5444], to be used (and shared) among MANET routing protocol security extensions, establishing two IANA registries for code-points for hash functions and cryptographic functions adhering to [RFC5444].

With respect to [RFC5444], this document:

- is intended to be used in the non-normative, but intended, mode of use of [RFC5444] as described in its Appendix B.

- is a specific example of the Security Considerations section of [RFC5444] (the authentication part).

5. Protocol Overview and Functioning

This specification does not describe a protocol, nor does it mandate specific router or protocol behavior. It represents a purely syntactical representation of security related information for use with [RFC5444] addresses, messages and packets, as well as establishes IANA registrations and registries.

6. Imported TLV Fields

In this specification, the following TLV fields from [RFC5444] are used:

<msg-hop-limit> - hop limit of a message, as specified in Section 5.2 of [RFC5444].

<msg-hop-count> - hop count of a message, as specified in Section 5.2 of [RFC5444].

<length> - length of a TLV in octets, as specified in Section 5.4.1 of [RFC5444].

7. General Signature TLV Structure

The following data structure allows representation of a cryptographic signature, including specification of the appropriate hash function and cryptographic function used for calculating the signature. This <signature> data structure is specified, using the regular expression syntax of [RFC5444], as:

```
<signature> := <hash-function>
               <cryptographic-function>
               <key-index>
               <signature-value>
```

where:

<hash-function> is an 8-bit unsigned integer field specifying the hash function.

<cryptographic-function> is an 8-bit unsigned integer field specifying the cryptographic function.

<key-index> is an 8-bit unsigned integer field specifying the key index of the key which was used to sign the message, which allows unique identification of different keys with the same originator. It is the responsibility of each key originator to make sure that actively used keys that it issues have distinct key indices and that all key indices have a value unequal to 0x00. Value 0x00 is reserved for a pre-installed, shared key.

<signature-value> is an unsigned integer field, whose length is <length> - 3, and which contains the cryptographic signature.

The basic version of this TLV assumes that calculating the signature can be decomposed into:

    signature-value = cryptographic-function(hash-function(content))

The hash function and the cryptographic function correspond to the entries in two IANA registries, set up by this specification in Section 12.

7.1. Rationale

The rationale for separating the hash function and the cryptographic function into two octets instead of having all combinations in a single octet - possibly as TLV type extension - is twofold: First, if further hash functions or cryptographic functions are added in the future, the number space might not remain continuous. More importantly, the number space of possible combinations would be rapidly exhausted. As new or improved cryptographic mechanism are continuously being developed and introduced, this format should be able to accommodate such for the foreseeable future.

The rationale for not including a field that lists parameters of the cryptographic signature in the TLV is, that before being able to validate a cryptographic signature, routers have to exchange or acquire keys (e.g. public keys). Any additional parameters can be provided together with the keys in that bootstrap process. It is therefore not necessary, and would even entail an extra overhead, to transmit the parameters within every message. One inherently included parameter is the length of the signature, which is <length> - 3 and which depends on the choice of the cryptographic function.

8. General Timestamp TLV Structure

The following data structure allows the representation of a timestamp. This <timestamp> data structure is specified as:

```
<timestamp> := <time-value>
```

where:

<time-value> is an unsigned integer field, whose length is <length>, and which contains the timestamp. The value of this variable is to be interpreted by the routing protocol as specified by the type extension of the Timestamp TLV, see Section 12.

A timestamp is essentially "freshness information". As such, its setting and interpretation is to be determined by the routing protocol (or the extension to a routing protocol) that uses it, and may e.g. correspond to a UNIX-timestamp, GPS timestamp or a simple sequence number.

9. Packet TLVs

Two Packet TLVs are defined, for including the cryptographic signature of a packet, and for including the timestamp indicating the time at which the cryptographic signature was calculated.

9.1. Packet SIGNATURE TLV

A Packet SIGNATURE TLV is an example of a Signature TLV as described in Section 7. When calculating the <signature-value> for a Packet, the signature is calculated over the three fields <hash-function>, <cryptographic-function> and <key-index> (in that order), concatenated with the entire Packet, including the packet header, all Packet TLVs (other than Packet SIGNATURE TLVs) and all included Messages and their message headers.

The following considerations apply:

- o As packets defined in [RFC5444] are never forwarded by routers, it is unnecessary to consider mutable fields (e.g. <msg-hop-count> and <msg-hop-limit>), if present, when calculating the signature.

- o any Packet SIGNATURE TLVs already present in the Packet TLV block MUST be removed before calculating the signature, and the Packet TLV block size MUST be recalculated accordingly. The TLVs can be restored after having calculated the signature value.

The rationale for removing any Packet SIGNATURE TLV already present prior to calculating the signature, is that several signatures may be added to the same packet, e.g., using different signature functions.

9.2. Packet TIMESTAMP TLV

A Packet TIMESTAMP TLV is an example of a Timestamp TLV as described in Section 8. If a packet contains a TIMESTAMP TLV and a SIGNATURE TLV, the TIMESTAMP TLV SHOULD be added to the packet before any SIGNATURE TLV, in order that it be included in the calculation of the signature.

10. Message TLVs

Two Message TLVs are defined, for including the cryptographic signature of a message, and for including the timestamp indicating the time at which the cryptographic signature was calculated.

10.1. Message SIGNATURE TLV

A Message SIGNATURE TLV is an example of a Signature TLV as described in Section 7. When determining the <signature-value> for a message, the signature is calculated over the three fields <hash-function>, <cryptographic-function>, and <key-index> (in that order), concatenated with the entire message with the following considerations:

- o the fields <msg-hop-limit> and <msg-hop-count>, if present, MUST both be assumed to have the value 0 (zero) when calculating the signature.

- o any Message SIGNATURE TLVs already present in the Message TLV block MUST be removed before calculating the signature, and the message size as well as the Message TLV block size MUST be recalculated accordingly. The TLVs can be restored after having calculated the signature value.

The rationale for removing any Message SIGNATURE TLV already present prior to calculating the signature, is that several signatures may be added to the same message, e.g., using different signature functions.

10.2. Message TIMESTAMP TLV

A Message TIMESTAMP TLV is an example of a Timestamp TLV as described in Section 8. If a message contains a TIMESTAMP TLV and a SIGNATURE TLV, the TIMESTAMP TLV SHOULD be added to the message before the SIGNATURE TLV, in order that it be included in the calculation of the signature.

11. Address Block TLVs

Two Address Block TLVs are defined, for associating a cryptographic signature to an address, and for including the timestamp indicating the time at which the cryptographic signature was calculated.

11.1. Address Block SIGNATURE TLV

An Address Block SIGNATURE TLV is an example of a Signature TLV as described in Section 7. The signature is calculated over the three fields <hash-function>, <cryptographic-function>, and <key-index> (in that order), concatenated with the address, concatenated with any other values, for example, any other TLV value that is associated with that address. A routing protocol or routing protocol extension using Address Block SIGNATURE TLVs MUST specify how to include any such concatenated attribute of the address in the verification process of the signature.

11.2. Address Block TIMESTAMP TLV

An Address Block TIMESTAMP TLV is an example of a Timestamp TLV as described in Section 8. If both a TIMESTAMP TLV and a SIGNATURE TLV are associated with an address, the timestamp value should be considered when calculating the value of the signature.

12. IANA Considerations

This section specifies requests to IANA.

12.1. TLV Registrations

This specification defines:

- o two Packet TLV types which must be allocated from the 0-223 range of the "Assigned Packet TLV Types" repository of [RFC5444] as specified in Table 1,

- o two Message TLV types which must be allocated from the 0-127 range of the "Assigned Message TLV Types" repository of [RFC5444] as specified in Table 2,

- o and two Address Block TLV types which must be allocated from the 0-127 range of the "Assigned Address Block TLV Types" repository of [RFC5444] as specified in Table 3.

This specification requests:

- set up of type extension registries for these TLV types.

IANA is requested to assign the same numerical value to the Packet TLV, Message TLV and Address Block TLV types with the same name.

12.1.1. Expert Review: Evaluation Guidelines

For the registries for TLV type extensions where an Expert Review is required, the designated expert SHOULD take the same general recommendations into consideration as are specified by [RFC5444].

For the Timestamp TLV, the same type extensions for all Packet, Message and Address TLVs should be numbered identically.

12.1.2. Packet TLV Type Registrations

The Packet TLVs as specified in Table 1 must be allocated from the "Packet TLV Types" namespace of [RFC5444].

Name	Type	Type Extension	Description
SIGNATURE	TBD3	0	Signature of a packet
		1-223	Expert Review
		224-255	Experimental Use
TIMESTAMP	TBD4	0	Unsigned timestamp of arbitrary length, given by the TLV length field. The MANET routing protocol has to define how to interpret this timestamp
		1-223	Expert Review
		224-255	Experimental Use

Table 1: Packet TLV types

12.1.3. Message TLV Type Registrations

The Message TLVs as specified in Table 2 must be allocated from the "Message TLV Types" namespace of [RFC5444].

Name	Type	Type Extension	Description
SIGNATURE	TBD1	0	Signature of a message
		1-223	Expert Review
		224-255	Experimental Use
TIMESTAMP	TBD2	0	Unsigned timestamp of arbitrary length, given by the TLV length field.
		1-223	Expert Review
		224-255	Experimental Use

Table 2: Message TLV types

12.1.4. Address Block TLV Type Registrations

The Address Block TLVs as specified in Table 3 must be allocated from the "Address Block TLV Types" namespace of [RFC5444].

Name	Type	Type Extension	Description
SIGNATURE	TBD1	0	Signature of an object (e.g. an address)
		1-223	Expert Review
		224-255	Experimental Use
TIMESTAMP	TBD2	0	Unsigned timestamp of arbitrary length, given by the TLV length field.
		1-223	Expert Review
		224-255	Experimental Use

Table 3: Address Block TLV types

12.2. New IANA Registries

This document introduces three namespaces that have been registered: Packet TLV Types, Message TLV Types, and Address Block TLV Types. This section specifies IANA registries for these namespaces and provides guidance to the Internet Assigned Numbers Authority regarding registrations in these namespaces.

The following terms are used with the meanings defined in [BCP26]: "Namespace", "Assigned Value", "Registration", "Unassigned", "Reserved", "Hierarchical Allocation", and "Designated Expert".

The following policies are used with the meanings defined in [BCP26]: "Private Use", "Expert Review", and "Standards Action".

12.2.1. Expert Review: Evaluation Guidelines

For the registries for the following tables where an Expert Review is required, the designated expert SHOULD take the same general recommendations into consideration as are specified by [RFC5444].

12.2.2. Hash Function

IANA is requested to create a new registry for the hash functions that can be used when creating a signature. The initial assignments and allocation policies are specified in Table 4.

Hash function value	Algorithm	Description
0	none	The "identity function": the hash value of an object is the object itself
1-223		Expert Review
224-255		Experimental Use

Table 4: Hash-Function registry

12.2.3. Cryptographic Algorithm

IANA is requested to create a new registry for the cryptographic function. Initial assignments and allocation policies are specified in Table 5.

Cryptographic function value	Algorithm	Description
0	none	The "identity function": the value of an encrypted hash is the hash itself
1-223		Expert Review
224-255		Experimental Use

Table 5: Cryptographic function registry

13. Security Considerations

This document does not specify a protocol itself. However, it provides a syntactical component for cryptographic signatures of messages and packets as defined in [RFC5444]. It can be used to address security issues of a protocol or extension that uses the component specified in this document. As such, it has the same security considerations as [RFC5444].

In addition, a protocol that includes this component MUST specify the usage as well as the security that is attained by the cryptographic signatures of a message or a packet.

As an example, a routing protocol that uses this component to reject "badly formed" messages if a control message does not contain a valid signature, should indicate the security assumption that if the signature is valid, the message is considered valid. It also should indicate the security issues that are counteracted by this measure (e.g. link or identity spoofing) as well as the issues that are not counteracted (e.g. compromised keys).

14. Acknowledgements

The authors would like to thank Jerome Milan (Ecole Polytechnique) for his advice as cryptographer. In addition, many thanks to Bo Berry (Cisco), Alan Cullen (BAE), Justin Dean (NRL), Christopher Dearlove (BAE), Paul Lambert (Marvell), and Henning Rogge (FGAN) for their constructive comments on the document.

15. References

15.1. Normative References

[BCP26] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, BCP 26, May 2008.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, BCP 14, March 1997.

[RFC5444] Clausen, T., Dearlove, C., Dean, J., and C. Adjih, "Generalized MANET Packet/Message Format", RFC 5444, February 2009.

15.2. Informative References

[NHDP] Clausen, T., Dean, J., and C. Dearlove, "MANET Neighborhood Discovery Protocol (NHDP)", RFC 6130, March 2011.

[OLSRv2] Clausen, T., Dearlove, C., and P. Jacquet, "The Optimized Link State Routing Protocol version 2", work in progress draft-ietf-manet-olsrv2-11.txt, April 2010.

Appendix A. Examples

A.1. Example of a Signed Message

The sample message depicted in Figure 1 is derived from the appendix of [RFC5444]. A SIGNATURE Message TLV has been added, with the value representing a 15 octet long signature of the whole message.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |0 0 0 0 1 0 0 0|    Packet Sequence Number     | Message Type  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |1 1 1 1 0 0 1 1|0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0|   Orig Addr   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           Originator Address (cont)           |   Hop Limit   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   Hop Count   |    Message Sequence Number    |0 0 0 0 0 0 0 0|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |0 0 1 1 1 1 1 0|   SIGNATURE   |0 0 0 1 0 0 0 0|0 0 0 1 0 0 1 0|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   Hash Func   |  Crypto Func  |   Key Index   | Sign. Value   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                    Signature Value (cont)                     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                    Signature Value (cont)                     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                    Signature Value (cont)                     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |    Signature Value (cont)     |   TLV Type    |0 0 0 1 0 0 0 0|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |0 0 0 0 0 1 1 0|                     Value                     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                 Value (cont)                  |0 0 0 0 0 0 1 0|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |0 0 1 1 0 0 0 0|0 0 0 0 0 0 1 0|              Mid              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |              Mid              | Prefix Length |0 0 0 0 0 0 0 0|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |0 0 0 0 0 0 0 0|0 0 0 0 0 0 1 1|1 0 0 0 0 0 0 0|0 0 0 0 0 0 1 0|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |             Head              |              Mid              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |              Mid              |              Mid              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1|   TLV Type    |0 0 0 1 0 0 0 0|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |0 0 0 0 0 0 1 0|             Value             |   TLV Type    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |0 0 1 0 0 0 0 0|  Index Start  |  Index Stop   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: Example message with signature

Authors' Addresses

Ulrich Herberg
LIX, Ecole Polytechnique
91128 Palaiseau Cedex,
France

Phone: +33 1 6933 4126
Email: ulrich@herberg.name
URI: http://www.herberg.name/

Thomas Heide Clausen
LIX, Ecole Polytechnique
91128 Palaiseau Cedex,
France

Phone: +33 6 6058 9349
Email: T.Clausen@computer.org
URI: http://www.thomasclausen.org/

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: August 21, 2011

R. Cole
US Army CERDEC
J. Macker
Naval Research Laboratory
A. Morton
AT&T Laboratories
February 17, 2011

Definition of Managed Objects for Performance Reporting
draft-ietf-manet-report-mib-01

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 objects for configuring autonomous report generation on any device that supports MIBs containing counter and gauge objects for performance monitoring. This allows a management station to instruct a device to build off-line reports to be collected asynchronously by the management station. Further, this REPORT-MIB can be configured in a proxy configuration where the report generation is performed on a device in close network proximity to the device containing the referenced counter objects. Hence, this capability allows network operators to reduce the SNMP polling traffic burden on Mobile Ad-Hoc and Disruption Tolerant Networks which is typical of SNMP performance management applications. This capability also improves the accuracy of the performance reports by minimizing the delay variation between the reporting agent (this MIB) and the data monitor (the MIB containing the monitored counter objects).

Status of This Memo

Table of Contents

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 configuring autonomous, off-line report generation for performance monitoring on any device supporting MIBs containing variables that resolve to type Integer32 (i.e., Integer32, Counter, Gauge, or TimeTicks). This REPORT-MIB allows for the report generation to occur on the same device as containing the referenced counter object or on a device in close network proximity to the device with the referenced counter object. This should be useful to devices or networks where efficient use of bandwidth is of concern or where intermittent connectivity is common. Hence, the REPORT-MIB is useful for devices managed over some Mobile Ad-Hoc Networks (MANETs) or Disruption Tolerant Networks (DTNs).

The REPORT-MIB offers three types of off-line reporting. One type offering reports which present statistical analysis of the objects being tracked; found within the reportStatsGroup. The second type offering a means to collect sampled data related to defined MIB objects. This second type of reporting is contained in the reportSampledGroup. The third offering reports which present (collect) raw data values and their time of change from the objects being tracked; found within the reportHistoryGroup.

For statistical reporting, the REPORT-MIB borrows from the RMON [RFC1757] ReportsControl and Reports Tables. Here the reportStatsCapabilitiesGroup defines the capabilities of the device with respect performance monitoring and statistical analysis. Some analysis is hard-coded into the definition of the reportStatsDataGroup while the device can also advertise extended statistical reporting via the reportMetricExtDefTable. The reportsControlTable specifies the report metrics, the Object ID to monitor and other aspects of the statistical report development and storage.

For the collection of sampled data, the REPORT-MIB draws directly from the usrHistoryGroup from RMON 2 [RFC2021]. Here the reportSampledControlTable allows the user to define aspects of the report for sampled data, including the number of MIB objects to be sampled and the nature of the sampling frequency and overall report duration. This group uses the notion of buckets, which contained sampled data from a set of identified MIB objects sampled at the same time point. The report consists of the buckets, each containing sets of sampled data from the selected MIB objects but at the specific sampling times. The reportSampledObjectTable allows the user to identify the multiple MIB objects to be sampled. The reportSampledDataTable contains the storage of the reported sampled data contained within buckets, one bucket for each time sampling instance.

For the collection of raw data, the REPORT-MIB contains a reportHistoryGroup comprised of the reportHistoryControlTable for control of historical data reports and the reportHistoryDataTable for the storage of the historical reports.

Various compliance groups are defined which allow for development of raw data collection reports, collection of sampled data reports or only statistical data reports, or all combinations.

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. Conventions

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 [RFC2119].

4. Overview

The REPORT-MIB references performance objects in other MIBs (and in other devices) and generates offline performance reports on those referenced objects. The REPORT-MIB can be coincident with the other MIB or can reside on another device in close network proximity to the device containing the referenced performance related object.

4.1. REPORT-MIB Management Model

This section describes the management model for the REPORT-MIB process. First, the model for the reportStatsGroup is presented. Then the models for the reportSampledGroup and the reportHistoryGroup are presented.

Figure 1 illustrates a potential use of the REPORT-MIB for the generation of off-line, remotely generated reports. The management station on the left hand side of the illustration instructs the remote device to create reports through manipulation of the ReportCntrl Objects in the REPORT-MIB resident on the remote device. The reports instruct the device to monitor the status of specified counters (on other MIBs and potentially on other devices in close network proximity) periodically and to generate a set of metrics describing the temporal behavior of those counter values. The reports are stored locally until the management station decides to pull them off the device. The figure shows a case where the REPORT-MIB generates a notification that Report_2 has completed, prompting the management station to pull Report_2 from the device.

```
   Mgmt       Device
   Station
                          REPORT-MIB         MIB_1
                          +---------+        +-----+
                          |         |        |     |
    Build_Rep_1           |+-------+|        +--+  |
       +----------------->||cntrl_1||<------>|PC|  |
                          |+-------+|<-+     +--+--+
                          |         |  |
                          |         |  |     MIB_2
                          |         |  |     +-----+
                          |         |  |     |     |
                          |         |  |     +--+  |
                          |         |  +---->|PC|  |
    Build_Rep_2           |+-------+|    +-->+--+--+
       +----------------->||cntrl_2||<---+
                          |+-------+|
                          |         |
                          |         |
    Build_Rep_3           |+-------+|
       +----------------->||cntrl_3||<---+   MIB_n
                          |+-------+|    |   +-----+
                          |         |    |   |     |
                          |+-----+  |    |   +--+  |
                          ||Rep_1|  |    +-->|PC|  |
                          |+-----+  |        +--+--+
                          |         |
                          |+-----+  |
       <------------------||Rep_2|  |
    Notf_Rep_2            |+-----+  |
                          |         |
                          |+-----+  |
                          ||Rep_3|  |
                          |+-----+  |
    Get_Rep_2             |         |
       +----------------->|         |
                          |         |
       <-----------------+|         |
    Send_Rep_2            +---------+
```

Figure 1: REPORT-MIB front-end report generation process.

The REPORT-MIB's reportStatsGroup defines specifically a set of metrics which are computed within all reports. It also allows for the specification of metric extensions which are local to the specific implementation of the REPORT-MIB. These are identified in the reportStatsCapabilitiesGroup metricExtension Table.

Each metric has an associated Object ID of type counter associated with it. The control table specifies a report interval and a bin interval. The report interval is an integral multiple of the bin interval. For each bin interval, the device identifies the change in the counter value over the bin interval (called x_i) and then computes the associated metric, e.g., sum, sum of the square, etc, over the set {x_i}. It maintains the sum of these computations within the metric objects in the 'reportStatsDataTable'. Once the report interval is complete, the management station has enough information to compute a set of interesting and useful statistics.

The computational model of the reportStatsGroup of the REPORT-MIB is illustrated in the figure below. The important controls are a) the contrlInterval, b) the cntrlBinInterval, c) the specific counterObjectId, and d) the metric. In the figure x_i represents the ith value of the counter change, i.e., x_i = counterValue(t_i+1) - counterValue(t_i). The metrics reported are then computed from the set (x_i). Three examples are identified in the figure, e.g., StatSumX, StatSumSq and StatMaxX. Other existing and potential metrics are discussed below.

```
 |
 |<-------------------- cntrlInterval --------------------------->|
 |                                                                 |

 |     |     |     |     |                             |     |     |
 | x_0 | x_1 | x_2 | x_3 |           ...               |x_n-2|x_n-1|
 +-----+-----+-----+-----+---                       ---+-----+-----+

 ^     ^     ^     ^     ^                             ^     ^     ^
 t_0   t_1   t_2   t_3   t_4                           t_n-2 t_n-1 t_n


 where t_i - t_i-1 = cntrlBinInterval
       n = cntrlInterval / cntrlBinInterval

       StatSumX = Sum(x_i)  from i=0, ..., n-1
       StatSumSq = Sum((x_i)^2) from i=0, ..., n-1
       StatMaxX = Max(x_i) for i=0, ..., n-1
```

Figure 2: REPORT-MIB statistical analysis computation process.

This capability then allows for the computation of various significant statistics related to the behavior of the referenced object.

- o Maximum and Minimum - the maximum and the minimum change in the referenced object during a single cntrlBinInterval during the cntrlInterval.

- o Arithmetic Mean - the mean change in the referenced object over all control bin intervals during the cntrlInterval. This is derived from the StatSumX quantity.

- o Variance - the variance in the change of the referenced object over all control bin intervals within the cntrlInterval. This is derived from the StatSumSq and the StatSumX quantities.

These are accessible from the statistical datum provided by this MIB module. Other statistics are derivable including, e.g., the slope of a least-squares fit to the rate of change of the referenced object. These are described below.

The REPORT-MIB also provides for the collection of sampled data instead of statistical data. It does this by importing (copying) the usrHistory group from RMON2 [RFC2021] which allows for the generation of reports collecting the sampled object values binned for the purpose of aggregation and efficiency of collection. These are defined within the reportSampledGroup. The model used for this type of report generation is based upon three tables. The reportSampledControlTable defines aspects of the report generation related to duration of the reporting interval, the bin (or bucket) sizes for the report, and the number of object values collected for each bucket. The reportUsrHistoryObjectTable identifies the specific MIB objects whose values are binned within the report. And the reportSampledDataTable contains the binned data values collected for the report.

The REPORT-MIB also provides for the collection of historical data instead of statistical or sampled data. It does this by defining the reportHistoryControlTable for the control of the historical reports and the reportHistoryDataTable for the storage of the historical reports.

4.2. Terms

The following definitions apply throughout this document:

- o Capabilities - Objects related to the capabilities of the device and MIB implemented on the device. Some objects are explicitly defined within the REPORT-MIB. Other capabilities can be exposed through the REPORT-MIB, but which are not explicitly defined within this document. These later capabilities include objects, e.g., for new metrics.

- o Control - Objects defined within this document which set the parameters for specific reports to be generated offline on the the remote managed device.

- o Data - Objects which hold the report data, either statistical, sampled or raw history data.

5. Structure of the MIB Module

This section presents the structure of the REPORT-MIB module. The objects are arranged into the following groups:

- o reportMIBNotifications - defines the notifications associated with the REPORT-MIB.

- o reportMIBObjects - defines the objects forming the basis for the REPORT MIB. These objects are divided up by function into the following groups:

- o

  * Statistics Group - This group contains the objects which support the generation of reports of a statistical nature.

  * Sampled Group - This group contains the objects which support the generation (collection) of reports exposing sampled data values.

  * History Group - This group contains the objects which support the generation (collection) of historical reports exposing raw data values.

- o reportMIBConformance - Defines a variety of conformance of implementations of this REPORT-MIB.

5.1. Textual Conventions

The textual conventions used in the REPORT-MIB are as follows. The RowStatus textual convention is imported from RFC 2579 [RFC2579].

5.2. The Statistics Group

The REPORT-MIB Statistics Group contains objects which allows for the generation of statistical analysis reports. For example, this group can be exercised to generate the mean and variance of the referenced counter object. The Statistics Group is composed of:

- o reportStatsCapabilitiesGroup - lists the statistics collections capabilities of this device. Certain statistics are mandatory, i.e., hard coded into the MIB definitions. While, the capabilities group allows the developer to add additional statistical analysis capabilities.

- o reportStatsControlGroup - allows the management application to define the parameters of the reports.

- o reportStatsDataGroup - presents the data from the specified reports.

As an example of how the metrics are to be computed within the REPORT-MIB, consider the standard metric object 'reportStatsDataStatSumX'. For each bin interval defined by the object reportCntrlReportsBinInterval, the change in the value of the counter pointed to by the Object ID reportCntrlReportsPriObjID is calculated. Then this (delta) value is added to the current value of the value contained in the object 'reportStatsDataStatSumX'. Then, if interested in computing the average change in this object (sampled each bin interval) for the duration of the report, the management station simply divides reportStatsDataStatSumX by reportStatsDataStatN. Although this is a trivial example because the value of reportAggrReportStatSumX is simple the difference in the counter reportCntrolReportsPriObjID at the start and the end of the total report interval, the other metrics defined are not as trivial.

The objects 'reportStatsDataOverflowStatSumX' and 'reportStatsDataHCSumX' are borrowed from RMON [RFC2021] and exist to handle integer overflow situations where, e.g., 'reportStatsDataStatSumX' overruns its maximum value numerous times.

Computation of the least-square fit of the data collected for a report can be accomplished. (NOTE: describe this capability here.)

5.3. The Sampled Group

The Sampled Group contains tables which allows for the development of reports based upon sampling the referenced counter objects at specified intervals. The development of this group within the REPORT-MIB follows exactly the User History group from the RMON 2 MIB [RFC2021]. The Sampled Group is composed of:

- o reportSampledControlTable - allows for the setting of the parameters of the report.

- o reportSampledObjectTable - sets the referenced objects to be sampled during the test. With this capability, the management application can reference multiple objects, all of which are sampled during the test and reported out through the reportSampledData Table.

- o reportSampledDataTable - contains the reports.

5.4. The History Group

The History Group contains tables which capture information on change events for the referenced objects. Depending upon the referenced objects, this could force the generation of large amounts of data. Care should be exercised when considering the use of this capability.

- o reportHistoryControlTable - defines the parameters for the test.

- o reportHistoryDataTable - presents the reports associated with the constructed tests.

5.5. The Notifications Group

The Notifications Sub-tree contains the list of notifications supported within the REPORT-MIB and their intended purpose or utility. (Note: This group is currently empty.)

6. Relationship to Other MIB Modules

[TODO]: The text of this section specifies the relationship of the MIB modules contained in this document to other standards, particularly to standards containing other MIB modules. Definitions imported from other MIB modules and other MIB modules that SHOULD be implemented in conjunction with the MIB module contained within this document are identified in this section.

6.1. Relationship to the SNMPv2-MIB

The 'system' group in the SNMPv2-MIB [RFC3418] is defined as being mandatory for all systems, and the objects apply to the entity as a whole. The 'system' group provides identification of the management entity and certain other system-wide data. The REPORT-MIB does not duplicate those objects.

6.2. Relationship to the RMON2-MIB

The REPORT-MIB is closely related in many aspects to the RMON2-MIB [RFC2021]. Specifically, the reportSampledGroup is a direct copy of the RMON2 User History Group, with the names changed to comply with the naming conventions within the REPORT-MIB. Further, the design and use of the control tables within the REPORT-MIB draw exactly from the definition of these table structures in the earlier RMON MIBs.

6.3. Relationship to the TPM-MIB

The REPORT-MIB pulled the reportStatsGroup directory from the TPM-MIB [RFC4150]. The table structures and the choice of statistics draws directly from the earlier TPM-MIB developed within the RMON Working Group.

6.4. MIB modules required for IMPORTS

[TODO]: Citations are not permitted within a MIB module, but any module mentioned in an IMPORTS clause or document mentioned in a REFERENCE clause is a Normative reference, and must be cited someplace within the narrative sections. If there are imported items in the MIB module, such as Textual Conventions, that are not already cited, they can be cited in text here. Since relationships to other MIB modules should be described in the narrative text, this section is typically used to cite modules from which Textual Conventions are imported.

The REPORT-MIB module IMPORTS objects from SNMPv2-SMI [RFC2578], SNMPv2-TC [RFC2579], SNMPv2-CONF [RFC2580], and IF-MIB [RFC2863]

7. Definitions

```
REPORT-MIB DEFINITIONS ::= BEGIN

IMPORTS

   ZeroBasedCounter32
      FROM RMON2-MIB                               -- [RFC2021]

   MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
   Counter32, Gauge32, Unsigned32, Integer32, mib-2
      FROM SNMPv2-SMI                              -- [RFC2578]

   TEXTUAL-CONVENTION, RowStatus,
   TimeStamp, StorageType
      FROM SNMPv2-TC                               -- [RFC2579]

   MODULE-COMPLIANCE, OBJECT-GROUP,
   NOTIFICATION-GROUP
      FROM SNMPv2-CONF                             -- [RFC2580]

   OwnerString
```

```
      FROM RMON-MIB                                -- [RFC2819]

   ZeroBasedCounter64
      FROM HCNUM-TC                                -- [RFC2856]

   SnmpAdminString
      FROM SNMP-FRAMEWORK-MIB                      -- [RFC3411]

   InetAddress, InetAddressType
      FROM INET-ADDRESS-MIB                        -- [RFC4001]

   SspmClockSource, SspmClockMaxSkew,
   SspmMicroSeconds
      FROM SSPM-MIB                               -- [RFC4149]
   ;

reportMIB MODULE-IDENTITY
   LAST-UPDATED "201102171300Z"  -- February 17, 2011
   ORGANIZATION "IETF MANET Working Group"
   CONTACT-INFO
      "WG E-Mail: manet@ietf.org

       WG Chairs: ian.chakeres@gmail.com
                  jmacker@nrl.navy.mil


       Editors:   Robert G. Cole
                  US Army CERDEC
                  328 Hopkins Road
                  Aberdeen Proving Ground, MD 21005
                  USA
                  +1 410 278-6779
                  robert.g.cole@us.army.mil

                  Joseph Macker
                  Naval Research Laboratory
                  Washington, D.C. 20375
                  USA
                  macker@itd.nrl.navy.mil

                  Al Morton
                  AT&T Laboratories
                  Middletown, N.J. 07724
                  USA
                  amorton@att.com"
   DESCRIPTION
      "This MIB module contains managed object definitions for
       the autonmous reporting of performance object counters.
```

```
       Copyright (C) The IETF Trust (2009). This version
       of this MIB module is part of RFC xxxx; see the RFC
       itself for full legal notices."

     -- Revision History
     REVISION    "201102171300Z"   -- February 17, 2011
     DESCRIPTION
        "The fifth draft of this MIB module published as
         draft-ietf-manet-report-mib-01.txt.  This document
         has been promoted to a MANET Working Group
         draft.

         Revisions to this draft include
         a) Proposed changes to the statsReport table to
            simplify communications between device and
            mgmt application,
         b) Added Notifications,
         c) Changed the reporting structure of the
            Sampled and the History reporting
            to align with the structure of the
            Statistics reports for the purpose of
            allowing for efficient notification and
            collection of data reports.
         d) Ran through smilint to clean up all errors
            and most warning.  A few still remain.
         "
     REVISION    "201007051300Z"   -- July 05, 2010
     DESCRIPTION
        "The fourth draft of this MIB module published as
         draft-ietf-manet-report-mib-00.txt.  This document
         has been promoted to a MANET Working Group
         draft.

         Significant revisions to this draft include
         a) added support for proxy configurations through
         the addition of address objects associated with
         the referenced counter objects associated with the
         performance reports."
     REVISION    "201003021300Z"   -- March 02, 2010
     DESCRIPTION
        "The third draft of this MIB module published as
         draft-cole-manet-report-mib-02.txt.  Significant
         revisions to this draft include a) changed naming
         of usrHistoryGroup to sampledGroup and  b) added
         a historyGroup."
     REVISION    "200910251300Z"   -- October 25, 2009
     DESCRIPTION
        "The second draft of this MIB module published as
```

```
         draft-cole-manet-report-mib-01.txt.  Significant
         revisions to this draft include a) the inclusion of
         raw data collection borrow blatently from the
         usrHistory Group within RMON2, b) the deletion of
         the CurrentHistoryTable from version -00,
         c) modifications to the overall structure of the
         MIB, and d) the definition of various Compliance
         options for implementations related to this MIB."
     REVISION    "200904281300Z"   -- April 28, 2009
     DESCRIPTION
        "Initial draft of this MIB module published as
         draft-cole-manet-report-mib-00.txt."
     -- RFC-Editor assigns XXXX
     ::= { mib-2 998 }   -- to be assigned by IANA



-- TEXTUAL CONVENTIONs

   ReportMetricDefID ::= TEXTUAL-CONVENTION
       DISPLAY-HINT "d"
       STATUS       current
       DESCRIPTION
           "An index that identifies through reference to a specific
           statistical metrics.
           "
       SYNTAX       Unsigned32 (1..2147483647)



--
-- Top-Level Object Identifier Assignments
--

reportMIBNotifications OBJECT IDENTIFIER ::= { reportMIB 0 }
reportMIBObjects       OBJECT IDENTIFIER ::= { reportMIB 1 }
reportMIBConformance   OBJECT IDENTIFIER ::= { reportMIB 2 }

-- The reportMIBObjects Assignments:
--      reportStatsGroup          - 1
--      reportSampledGroup        - 2
--      reportHistoryGroup        - 3




reportStatsGroup       OBJECT IDENTIFIER ::= { reportMIBObjects 1 }
```

```
-- Then, the reportStatsGroup assignments are :
--      reportStatsCapabilitiesGroup    - 1
--      reportStatsControlGroup         - 2
--      reportStatsDataGroup            - 3


-- reportStatsCapabilitiesGroup
--    This group contains the REPORT objects that identify specific
--    capabilities within this device related to REPORT functions.


reportCapabilitiesGroup  OBJECT IDENTIFIER ::= { reportStatsGroup 1 }

reportClockResolution  OBJECT-TYPE
    SYNTAX      SspmMicroSeconds
    MAX-ACCESS  read-only
    STATUS      current
    -- UNITS       Microseconds
    DESCRIPTION
        "A read-only variable indicating the resolution
         of the measurements possible by this device."
    ::= { reportCapabilitiesGroup 1 }

reportClockMaxSkew  OBJECT-TYPE
    SYNTAX      SspmClockMaxSkew
    MAX-ACCESS  read-only
    STATUS      current
    -- UNITS       Seconds
    DESCRIPTION
        "A read-only variable indicating the maximum
         offset error due to skew of the local clock
         over the time interval 86400 seconds, in seconds."
    ::= { reportCapabilitiesGroup 2 }

reportClockSource  OBJECT-TYPE
    SYNTAX      SspmClockSource
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "A read-only variable indicating the source of the clock.
         This is provided to allow a user to determine how accurate
         the timing mechanism is compared with other devices."
    ::= { reportCapabilitiesGroup 3 }

reportMetricDirLastChange  OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
```

```
    DESCRIPTION
        "The value of sysUpTime at the time the
        reportTransMetricDirTable was last modified, through
        modifications of the reportTransMetricDirConfig object."
    ::= { reportCapabilitiesGroup 4 }


-- REPORT Metric Extensions Definition Table

reportMetricExtDefTable  OBJECT-TYPE
    SYNTAX      SEQUENCE OF ReportMetricExtDefEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The reportMetricExtDefTable describes the metrics
        available to the REPORT-MIB.  The reportMetricExtDefTable
        can define metrics by referencing existing IETF,
        ITU, and other standards organizations' documents,
        including enterprise-specific documents.
        Examples of appropriate references include the
        ITU-T Recommendation Y.1540 [Y.1540] on IP
        packet transfer performance metrics and the
        IETF documents from the IPPM WG; e.g., RFC2681
        on the round trip delay metric [RFC2681] or
        RFC3393 on the delay variation metric [RFC3393].
        Other examples include RFC2679 [RFC2679], RFC2680
        [RFC2680], and RFC3432 [RFC3432].  Although no
        specific metric is mandatory, implementations
        should, at a minimum, support a round-trip delay
        and a round-trip loss metric.

        This table contains one row per metric supported by this
        agent, and it should be populated during system
        initialization."
    ::= { reportCapabilitiesGroup 5 }

reportMetricExtDefEntry  OBJECT-TYPE
    SYNTAX      ReportMetricExtDefEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Information about a particular metric."
    INDEX   { reportMetricExtDefID }
    ::= { reportMetricExtDefTable 1 }

ReportMetricExtDefEntry ::= SEQUENCE {
      reportMetricExtDefID             ReportMetricDefID,
      reportMetricExtDefType           INTEGER,
```

```
      reportMetricExtDefName           SnmpAdminString,
      reportMetricExtDefOperation      SnmpAdminString,
      reportMetricExtDefReference      SnmpAdminString
    }

reportMetricExtDefID OBJECT-TYPE
    SYNTAX      ReportMetricDefID
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The index for this entry.  This object identifies
         the particular metric in this MIB module."
    ::= { reportMetricExtDefEntry 1 }

reportMetricExtDefType  OBJECT-TYPE
    SYNTAX      INTEGER  {
                          other(1),
                          singleObjMetric(2),
                          multipleObjMetric(3)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The basic type of metric indicated by this entry.

        The value 'other(1)' indicates that this metric cannot be
        characterized by any of the remaining enumerations specified
        for this object.

        The value 'connectMetric(2)' indicates that this metric
        measures connectivity characteristics.

        The value 'delayMetric(3)' indicates that this metric
        measures delay characteristics.
        "
    ::= { reportMetricExtDefEntry 2 }

reportMetricExtDefName  OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The textual name of this metric.  For example, if
        this reportMetricDefEntry identified the IPPM metric for
        round trip delay, then this object should contain
        the value, e.g., 'Type-P-Round-Trip-Delay'."
    ::= { reportMetricExtDefEntry 3 }
```

```
reportMetricExtDefOperation  OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The textual description of the operations necessary
        to compute this metric.  For example, if
        this reportMetricDefEntry identified the IPPM metric for
        round trip delay, then this object should contain
        the value, e.g., 'Type-P-Round-Trip-Delay'."
    ::= { reportMetricExtDefEntry 4 }

reportMetricExtDefReference  OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object contains a reference to the document that
        defines this metric.  If this document is available online
        via electronic download, then a de-referencable URL
        should be specified in this object.  The implementation
        must support an HTTP URL type and may support additional
        types of de-referencable URLs such as an FTP type.

        For example, if this reportMetricDefName identified the IPPM
        metric 'Type-P-Round-Trip-Delay', then this object should
        contain the value, e.g.,
        'http://www.ietf.org/rfc/rfc2681.txt'."
    ::= { reportMetricExtDefEntry 5 }



-- Stats Control Group
--     This and the following tables are modeled
--     after the report control and collection
--     capabilities found in RMON 2, RFC 2021

reportStatsControlGroup OBJECT IDENTIFIER ::= {reportStatsGroup 2}

reportStatsControlTable  OBJECT-TYPE
    SYNTAX      SEQUENCE OF ReportStatsControlEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The reportStatsControlTable is the controlling entry
        that manages the population of studies in the
        Report for selected time intervals.
```

```
        Note that this is not like the typical RMON
        controlTable and dataTable in which each entry creates
        its own data table.  Each entry in this table enables the
        creation of multiple data tables on a study basis.  For each
        interval, the study is updated in place, and the current
        data content of the table becomes invalid.

        The control table entries are persistent across
        system reboots."
    ::= { reportStatsControlGroup 1 }

reportStatsControlEntry  OBJECT-TYPE
    SYNTAX      ReportStatsControlEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A conceptual row in the reportStatsControlTable.

        An example of the indexing of this entry is
        reportGenReportCntrInterval.1"
    INDEX { reportStatsControlIndex }
    ::= { reportStatsControlTable 1 }

ReportStatsControlEntry ::= SEQUENCE {
    reportStatsControlIndex             Unsigned32,
    reportStatsControlInterval          Unsigned32,
    reportStatsControlBinInterval       Unsigned32,
    reportStatsControlPriObjID          OBJECT IDENTIFIER,
    reportStatsControlPriObjIpAddrType  InetAddressType,
    reportStatsControlPriObjIPAddr      InetAddress,
    reportStatsControlSecObj1ID         OBJECT IDENTIFIER,
    reportStatsControlSecObj1IpAddrType InetAddressType,
    reportStatsControlSecObj1IPAddr     InetAddress,
    reportStatsControlSecObj2ID         OBJECT IDENTIFIER,
    reportStatsControlSecObj2IpAddrType InetAddressType,
    reportStatsControlSecObj2IPAddr     InetAddress,
    reportStatsControlSecObj3ID         OBJECT IDENTIFIER,
    reportStatsControlSecObj3IpAddrType InetAddressType,
    reportStatsControlSecObj3IPAddr     InetAddress,
    reportStatsControlSecObj4ID         OBJECT IDENTIFIER,
    reportStatsControlSecObj4IpAddrType InetAddressType,
    reportStatsControlSecObj4IPAddr     InetAddress,
    reportStatsControlSecObj5ID         OBJECT IDENTIFIER,
    reportStatsControlSecObj5IpAddrType InetAddressType,
    reportStatsControlSecObj5IPAddr     InetAddress,
    reportStatsControlMetricExt1        ReportMetricDefID,
    reportStatsControlMetricExt2        ReportMetricDefID,
    reportStatsControlMetricExt3        ReportMetricDefID,
```

```
    reportStatsControlMetricExt4        ReportMetricDefID,
    reportStatsControlMetricExt5        ReportMetricDefID,
    reportStatsControlReqReports        Unsigned32,
    reportStatsControlGrantedReports    Unsigned32,
    reportStatsControlStartTime         TimeStamp,
    reportStatsControlReportNumber      Unsigned32,
    reportStatsControlInsertsDenied     Counter32,
    reportStatsControlOwner             OwnerString,
    reportStatsControlStorageType       StorageType,
    reportStatsControlStatus            RowStatus
}

reportStatsControlIndex  OBJECT-TYPE
    SYNTAX      Unsigned32 (1..65535)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "An index that uniquely identifies an entry in the
        reportStatsControlTable.  Each such entry defines a unique
        report whose results are placed in the reportGenReportTable
        on behalf of this reportStatsControlEntry."
    ::= { reportStatsControlEntry 1 }


reportStatsControlInterval  OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "Seconds"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
       "The interval in seconds over which data is accumulated before
        being aggregated into a report in the reportGenReportTable.
        All reports with the same reportStatsControlIndex will be
        based on the same interval.

        The value of the reportStatsControlInterval should be
        an integral multiple of the value of the
        reportStatsControlBinInterval.

        This object may not be modified if the associated
        reportStatsControlStatus object is equal to active(1)."
    DEFVAL { 3600 }
    ::= { reportStatsControlEntry 2 }

reportStatsControlBinInterval  OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "Seconds"
    MAX-ACCESS  read-create
```

```
    STATUS      current
    DESCRIPTION
       "The interval in seconds between which the value of the
        reportStatsControlPriObjID and SecObjIDs are polled
        for the purpose of generating the metric values associated
        with this report.  All reports with the same
        reportStatsControlIndex will be based on the
        same bin interval.

        This object may not be modified if the associated
        reportStatsControlStatus object is equal to active(1)."
    DEFVAL { 3600 }
    ::= { reportStatsControlEntry 3 }

reportStatsControlPriObjID  OBJECT-TYPE
    SYNTAX      OBJECT IDENTIFIER
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This identifies the primary counter object to be
        monitored within this report.

        This object may not be modified if the associated
        reportStatsControlStatus object is equal to active(1)."
    ::= { reportStatsControlEntry 4 }

reportStatsControlPriObjIpAddrType  OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This identifies the IP address type
        of the IP address associated with the
        primary counter object to be
        monitored within this report.

        This object may not be modified if the associated
        reportStatsControlStatus object is equal to active(1)."
    ::= { reportStatsControlEntry 5 }

reportStatsControlPriObjIPAddr  OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This identifies the IP addree of the
        primary counter object to be
        monitored within this report.
```

```
        This object may not be modified if the associated
        reportStatsControlStatus object is equal to active(1)."
    ::= { reportStatsControlEntry 6 }

reportStatsControlSecObj1ID  OBJECT-TYPE
    SYNTAX      OBJECT IDENTIFIER
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This identifies the secondary counter object to be
        monitored within this report associated with the
        specified reportStatsControlMetricExt1.  If the
        reportStatsControlMetricExt1 is a simple metric, then
        the value of this reportStatsControlSecObj1ID is
        set to '0'.

        This object may not be modified if the associated
        reportStatsControlStatus object is equal to active(1)."
    ::= { reportStatsControlEntry 7 }

reportStatsControlSecObj1IpAddrType  OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This identifies the IP address type
        of the IP address associated with the
        secondary counter object to be
        monitored within this report.

        This object may not be modified if the associated
        reportStatsControlStatus object is equal to active(1)."
    ::= { reportStatsControlEntry 8 }

reportStatsControlSecObj1IPAddr  OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This identifies the IP addree of the
        secondary counter object to be
        monitored within this report.

        This object may not be modified if the associated
        reportStatsControlStatus object is equal to active(1)."
    ::= { reportStatsControlEntry 9 }

reportStatsControlSecObj2ID  OBJECT-TYPE
```

```
    SYNTAX      OBJECT IDENTIFIER
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This identifies the secondary counter object to be
        monitored within this report associated with the
        specified reportStatsControlMetricExt2.  If the
        reportStatsControlMetricExt2 is a simple metric, then
        the value of this reportStatsControlSecObj2ID is
        set to '0'.

        This object may not be modified if the associated
        reportStatsControlStatus object is equal to active(1)."
    ::= { reportStatsControlEntry 10 }

reportStatsControlSecObj2IpAddrType  OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This identifies the IP address type
        of the IP address associated with the
        secondary counter object to be
        monitored within this report.

        This object may not be modified if the associated
        reportStatsControlStatus object is equal to active(1)."
    ::= { reportStatsControlEntry 11 }

reportStatsControlSecObj2IPAddr  OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This identifies the IP addree of the
        secondary counter object to be
        monitored within this report.

        This object may not be modified if the associated
        reportStatsControlStatus object is equal to active(1)."
    ::= { reportStatsControlEntry 12 }

reportStatsControlSecObj3ID  OBJECT-TYPE
    SYNTAX      OBJECT IDENTIFIER
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This identifies the secondary counter object to be
```

```
        monitored within this report associated with the
        specified reportStatsControlMetricExt3.  If the
        reportStatsControlMetricExt3 is a simple metric, then
        the value of this reportStatsControlSecObj3ID is
        set to '0'.

        This object may not be modified if the associated
        reportStatsControlStatus object is equal to active(1)."
    ::= { reportStatsControlEntry 13 }

reportStatsControlSecObj3IpAddrType  OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This identifies the IP address type
        of the IP address associated with the
        secondary counter object to be
        monitored within this report.

        This object may not be modified if the associated
        reportStatsControlStatus object is equal to active(1)."
    ::= { reportStatsControlEntry 14 }

reportStatsControlSecObj3IPAddr  OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This identifies the IP addree of the
        secondary counter object to be
        monitored within this report.

        This object may not be modified if the associated
        reportStatsControlStatus object is equal to active(1)."
    ::= { reportStatsControlEntry 15 }

reportStatsControlSecObj4ID  OBJECT-TYPE
    SYNTAX      OBJECT IDENTIFIER
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This identifies the secondary counter object to be
        monitored within this report associated with the
        specified reportStatsControlMetricExt4.  If the
        reportStatsControlMetricExt4 is a simple metric, then
        the value of this reportStatsControlSecObj4ID is
        set to '0'.
```

```
        This object may not be modified if the associated
        reportStatsControlStatus object is equal to active(1)."
    ::= { reportStatsControlEntry 16 }

reportStatsControlSecObj4IpAddrType  OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This identifies the IP address type
        of the IP address associated with the
        secondary counter object to be
        monitored within this report.

        This object may not be modified if the associated
        reportStatsControlStatus object is equal to active(1)."
    ::= { reportStatsControlEntry 17 }

reportStatsControlSecObj4IPAddr  OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This identifies the IP addree of the
        secondary counter object to be
        monitored within this report.

        This object may not be modified if the associated
        reportStatsControlStatus object is equal to active(1)."
    ::= { reportStatsControlEntry 18 }

reportStatsControlSecObj5ID  OBJECT-TYPE
    SYNTAX      OBJECT IDENTIFIER
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This identifies the secondary counter object to be
        monitored within this report associated with the
        specified reportStatsControlMetricExt5.  If the
        reportStatsControlMetricExt5 is a simple metric, then
        the value of this reportStatsControlSecObj5ID is
        set to '0'.

        This object may not be modified if the associated
        reportStatsControlStatus object is equal to active(1)."
    ::= { reportStatsControlEntry 19 }

reportStatsControlSecObj5IpAddrType  OBJECT-TYPE
```

```
    SYNTAX      InetAddressType
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This identifies the IP address type
        of the IP address associated with the
        secondary counter object to be
        monitored within this report.

        This object may not be modified if the associated
        reportStatsControlStatus object is equal to active(1)."
    ::= { reportStatsControlEntry 20 }

reportStatsControlSecObj5IPAddr  OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This identifies the IP addree of the
        secondary counter object to be
        monitored within this report.

        This object may not be modified if the associated
        reportStatsControlStatus object is equal to active(1)."
    ::= { reportStatsControlEntry 21 }

reportStatsControlMetricExt1  OBJECT-TYPE
    SYNTAX      ReportMetricDefID
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This identifies the first metric extension placed
        in the reportGenReportTable.  If no metric extension
        is requested, then this object value is set to '0'.

        If this metric is defined on a single counter object,
        then only the reportStatsControlPriObjID is set, while
        the value of the reportStatsControlSecObjID is
        set to '0'.  Else, the reportStatsControlSecObjID
        is set in accoradance with the instruction in the
        definition of the metric extension found in the
        reportCapabilitiesMetwircExtTable above.

        This object may not be modified if the associated
        reportStatsControlStatus object is equal to active(1)."
    ::= { reportStatsControlEntry 22 }

reportStatsControlMetricExt2  OBJECT-TYPE
```

```
    SYNTAX      ReportMetricDefID
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This identifies the second metric extension placed
        in the reportGenReportTable.  If no metric extension
        is requested, then this object value is set to '0'.

        If this metric is defined on a single counter object,
        then only the reportStatsControlPriObjID is set, while
        the value of the reportStatsControlSecObjID is
        set to '0'.  Else, the reportStatsControlSecObjID
        is set in accoradance with the instruction in the
        definition of the metric extension found in the
        reportCapabilitiesMetwircExtTable above.

        This object may not be modified if the associated
        reportStatsControlStatus object is equal to active(1)."
    ::= { reportStatsControlEntry 23 }

reportStatsControlMetricExt3  OBJECT-TYPE
    SYNTAX      ReportMetricDefID
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This identifies the third metric extension placed
        in the reportGenReportTable.  If no metric extension
        is requested, then this object value is set to '0'.

        If this metric is defined on a single counter object,
        then only the reportStatsControlPriObjID is set, while
        the value of the reportStatsControlSecObjID is
        set to '0'.  Else, the reportStatsControlSecObjID
        is set in accoradance with the instruction in the
        definition of the metric extension found in the
        reportCapabilitiesMetwircExtTable above.

        This object may not be modified if the associated
        reportStatsControlStatus object is equal to active(1)."
    ::= { reportStatsControlEntry 24 }

reportStatsControlMetricExt4  OBJECT-TYPE
    SYNTAX      ReportMetricDefID
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This identifies the fourth metric extension placed
        in the reportGenReportTable.  If no metric extension
```

```
        is requested, then this object value is set to '0'.

        If this metric is defined on a single counter object,
        then only the reportStatsControlPriObjID is set, while
        the value of the reportStatsControlSecObjID is
        set to '0'.  Else, the reportStatsControlSecObjID
        is set in accoradance with the instruction in the
        definition of the metric extension found in the
        reportCapabilitiesMetwircExtTable above.

        This object may not be modified if the associated
        reportStatsControlStatus object is equal to active(1)."
    ::= { reportStatsControlEntry 25 }

reportStatsControlMetricExt5  OBJECT-TYPE
    SYNTAX      ReportMetricDefID
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This identifies the fifth metric extension placed
        in the reportGenReportTable.  If no metric extension
        is requested, then this object value is set to '0'.

        If this metric is defined on a single counter object,
        then only the reportStatsControlPriObjID is set, while
        the value of the reportStatsControlSecObjID is
        set to '0'.  Else, the reportStatsControlSecObjID
        is set in accoradance with the instruction in the
        definition of the metric extension found in the
        reportCapabilitiesMetwircExtTable above.

        This object may not be modified if the associated
        reportStatsControlStatus object is equal to active(1)."
    ::= { reportStatsControlEntry 26 }


reportStatsControlReqReports  OBJECT-TYPE
    SYNTAX      Unsigned32 (1..65535)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The number of saved reports requested to be allocated on
        behalf of this entry.

        This object may not be modified if the associated
        reportStatsControlStatus object is equal to active(1)."
    ::= { reportStatsControlEntry 27 }
```

```
reportStatsControlGrantedReports  OBJECT-TYPE
    SYNTAX      Unsigned32 (0..65535)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of saved reports the agent has allocated based
        on the requested amount in reportStatsControlReqReports.
        Because each report can have many entries, the total number
        of entries allocated will be this number multiplied by the
        value of reportStatsControlGrantedSize, or by 1 if that
        object doesn't exist.

        When the associated reportStatsControlReqReports object is
        created or modified, the agent should set this object as
        closely to the requested value as is possible for the
        particular implementation and available resources.  When
        considering available resources, the agent must consider its
        ability to allocate this many reports, each with the number
        of entries represented by reportStatsControlGrantedSize, or
        by 1 if that object doesn't exist.

        Note that although the storage required for each report may
        fluctuate due to changing conditions, the agent must continue
        to have storage available to satisfy the full report size for
        all reports, when necessary.  Further, the agent must not
        lower this value except as a result of a set to the
        associated reportStatsControlReqSize object."
    ::= { reportStatsControlEntry 28 }

reportStatsControlStartTime  OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The value of sysUpTime when the system began processing the
        report in progress.  Note that the report in progress is not
        available.

        This object may be used by the management station to figure
        out the start time for all previous reports saved for this
        reportStatsControlEntry, as reports are started at fixed
        intervals."
    ::= { reportStatsControlEntry 29 }

reportStatsControlReportNumber  OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current
```

```
    DESCRIPTION
        "The number of the report in progress.  When an
        reportStatsControlEntry is activated, the first report will
        be numbered zero."
    ::= { reportStatsControlEntry 30 }

reportStatsControlInsertsDenied  OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of attempts to add an entry to reports for
        this ReportStatsControlEntry that failed because the
        number of entries would have exceeded
        reportStatsControlGrantedSize.

        This number is valuable in determining if enough
        entries have been allocated for reports
        in light of fluctuating network
        usage.  Note that an entry that is denied will
        often be attempted again, so this number will
        not predict the exact number of additional entries
        needed, but it can be used to
        understand the relative magnitude of the problem.

        Also note that there is no ordering specified for
        the entries in the report;
        thus, there are no rules for which entries
        will be omitted when not enough entries are available.
        As a consequence, the agent is not required
        to delete 'least valuable' entries first."
    ::= { reportStatsControlEntry 31 }

reportStatsControlOwner  OBJECT-TYPE
    SYNTAX      OwnerString
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The entity that configured this entry and is
        therefore using the resources assigned to it.

        This object may not be modified if the associated
        reportStatsControlStatus object is equal to active(1)."
    ::= { reportStatsControlEntry 32 }

reportStatsControlStorageType  OBJECT-TYPE
    SYNTAX      StorageType
    MAX-ACCESS  read-create
```

```
    STATUS      current
    DESCRIPTION
        "The storage type of this reportStatsControlEntry.  If the
        value of this object is 'permanent', no objects in this row
        need to be writable."
    ::= { reportStatsControlEntry 33 }

reportStatsControlStatus  OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The status of this performance control entry.

        An entry may not exist in the active state unless each
        object in the entry has an appropriate value.

        Once this object is set to active(1), no objects in the
        reportStatsControlTable can be changed.

        If this object is not equal to active(1), all associated
        entries in the reportGenReportTable shall be deleted."
    ::= { reportStatsControlEntry 34 }



-- Stats Data Group

reportStatsDataGroup  OBJECT IDENTIFIER ::= { reportStatsGroup 3 }


-- Report Stats Data Table

reportStatsDataTable  OBJECT-TYPE
    SYNTAX      SEQUENCE OF ReportStatsDataEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table contains completed
        studies for each of the control table entries in
        reportAggrReportCntrlTable.  These studies are
        provided based on the selections and parameters
        found for the entry in the
        reportAggregateReportsCntrlTable.

        The performance statistics are specified in the
        reportTransMetricDirTable associated with the
```

```
        application in question and indexed by
        appLocalIndex and reportTransMetricIndex."
    ::= { reportStatsDataGroup 1 }

reportStatsDataEntry  OBJECT-TYPE
    SYNTAX      ReportStatsDataEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A conceptual row in the reportStatsDataTable.

        The reportStatsControlIndex value in the
        index identifies the reportStatsControlEntry
        on whose behalf this entry was created.

        The reportStatsDataIndex value in the index
        identifies which report
        (in the series of reports) this entry is a part of.

        The reportStatsDataServerAddress value in the
        index identifies the network layer address of the
        device generatung this report.

        An example of the indexing of this entry is
        reportStatsDataStatN.3.15.34.262.18.4.128.2.6.7.3256521"
    INDEX { reportStatsControlIndex,
            reportStatsDataIndex
          }
    ::= { reportStatsDataTable 1 }

-- Note: Thinking about restructuring this
--    table somewhat, in order
--    to allow for a more complete report information to
--    simplify report collection from the remote
--    mgmt application.  Indicating below potential
--    additional objects.
ReportStatsDataEntry ::= SEQUENCE {
    reportStatsDataIndex                Unsigned32,
    -- reportStatsDataServerAddrType    inetAddressType,
    -- reportStatsDataServerAddress     inetAddress,
    reportStatsDataServerAddress        OCTET STRING,
    -- reportStatsDataReportStartTime   TimeStamp,
    -- reportStatsDataReportInterval    Unsigned32,
    reportStatsDataStatN                ZeroBasedCounter32,
    reportStatsDataStatSumX             ZeroBasedCounter32,
    reportStatsDataOverflowStatSumX     ZeroBasedCounter32,
    reportStatsDataHCStatSumX           ZeroBasedCounter64,
    reportStatsDataStatMaximum          ZeroBasedCounter32,
```

```
    reportStatsDataStatMinimum          ZeroBasedCounter32,
    reportStatsDataStatSumSq            ZeroBasedCounter32,
    reportStatsDataOverflowStatSumSq    ZeroBasedCounter32,
    reportStatsDataHCStatSumSq          ZeroBasedCounter64,
    reportStatsDataStatSumIX            ZeroBasedCounter32,
    reportStatsDataOverflowStatSumIX    ZeroBasedCounter32,
    reportStatsDataHCStatSumIX          ZeroBasedCounter64,
    reportStatsDataStatSumIXSq          ZeroBasedCounter32,
    reportStatsDataOverflowStatSumIXSq  ZeroBasedCounter32,
    reportStatsDataHCStatSumIXSq        ZeroBasedCounter64,
    reportStatsDataStatMetricExt1       ZeroBasedCounter32,
    reportStatsDataStatMetricExt2       ZeroBasedCounter32,
    reportStatsDataStatMetricExt3       ZeroBasedCounter32,
    reportStatsDataStatMetricExt4       ZeroBasedCounter32,
    reportStatsDataStatMetricExt5       ZeroBasedCounter32
}

reportStatsDataIndex  OBJECT-TYPE
    SYNTAX      Unsigned32 (1..2147483647)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The value of reportStatsControlReportNumber for the report to
        which this entry belongs."
    ::= { reportStatsDataEntry 1 }

-- [Note: Need to revisit the syntax for this object of type 'address'.]
reportStatsDataServerAddress  OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE (0..108))
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The network layer address of the server host in this
        conversation.

        This is represented as an octet string with specific
        semantics and length as identified by the
        protocolDirLocalIndex component of the index.

        Because this object is an index variable, it is encoded in
        the index according to the index encoding rules.  For
        example, if the protocolDirLocalIndex indicates an
        encapsulation of IPv4, this object is encoded as a length
        octet of 4, followed by the 4 octets of the IPv4 address,
        in network byte order.

        If the associated reportAggrReportCntrlAggrType is equal to
        application(4) or client(2), then this object will be a null
```

```
        string and will be encoded simply as a length octet of 0."
    ::= { reportStatsDataEntry 2 }

reportStatsDataStatN  OBJECT-TYPE
    SYNTAX      ZeroBasedCounter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The count of the total number of data points for the
        specified metric.  This number is simply the value of
        reportCntrlReportsInterval divided by the value of
        reportCntrlReportsBinInterval, which should be integer
        valued.
        "
    ::= { reportStatsDataEntry 3 }

reportStatsDataStatSumX  OBJECT-TYPE
    SYNTAX      ZeroBasedCounter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The sum of all the data point values for the specified
        metric.  This number always represents the total values
        of the statistical datum analyzed.  Each metric
        specifies the exact meaning of this object.
        This value represents the results of one metric and is
        related directly to the specific parameters of the metric
        and the Server and Client addresses involved."
    ::= { reportStatsDataEntry 4 }

reportStatsDataOverflowStatSumX  OBJECT-TYPE
    SYNTAX      ZeroBasedCounter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of times the associated
        reportAggrReportStatSumX counter has overflowed.
        Note that this object will only be instantiated if the
        associated reportAggrReportHCStatSumX object is also
        instantiated for a particular dataSource."
    ::= { reportStatsDataEntry 5 }

reportStatsDataHCStatSumX  OBJECT-TYPE
    SYNTAX      ZeroBasedCounter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The high-capacity version of reportAggrReportStatSumX.
```

```
        Note that this object will only be instantiated if the
        agent supports High Capacity monitoring for a particular
        dataSource."
    ::= { reportStatsDataEntry 6 }

reportStatsDataStatMaximum  OBJECT-TYPE
    SYNTAX      ZeroBasedCounter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The single maximum data point value observed during the
        study period for the specified metric.  This number always
        represents the maximum value of any single statistical
        datum analyzed.  Each metric specifies the exact meaning
        of this object.

        This value represents the results of one metric and is
        related directly to the specific parameters of the metric
        and the Server and Client addresses involved."
    ::= { reportStatsDataEntry 7 }

reportStatsDataStatMinimum  OBJECT-TYPE
    SYNTAX      ZeroBasedCounter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The single minimum data point value observed during the
        study period for the specified metric.  This number always
        represents the minimum value of any single statistical
        datum analyzed.  Each metric specifies the exact meaning
        of this object.

        This value represents the results of one metric and is
        related directly to the specific parameters of the metric
        and the Server and Client addresses involved."
    ::= { reportStatsDataEntry 8 }

reportStatsDataStatSumSq  OBJECT-TYPE
    SYNTAX      ZeroBasedCounter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The sum of all the squared data point values for the
        specified metric.  This number always represents the
        total of the squared values of the statistical datum
        analyzed.  Each metric specifies the exact meaning of
        this object.
```

```
        This value represents the results of one metric and is
        related directly to the specific parameters of the metric
        and the Server and Client addresses involved."
    ::= { reportStatsDataEntry 9 }

reportStatsDataOverflowStatSumSq  OBJECT-TYPE
    SYNTAX      ZeroBasedCounter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of times the associated
        reportAggrReportStatSumSq counter has overflowed.
        Note that this object will only  be instantiated if
        the associated reportAggrReportHCStatSumSq object
        is also instantiated for a particular dataSource."
    ::= { reportStatsDataEntry 10 }

reportStatsDataHCStatSumSq  OBJECT-TYPE
    SYNTAX      ZeroBasedCounter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The high-capacity version of reportAggrReportStatSumSq.
        Note that this object will only be instantiated if the
        agent supports High Capacity monitoring for a particular
        dataSource."
    ::= { reportStatsDataEntry 11 }

reportStatsDataStatSumIX  OBJECT-TYPE
    SYNTAX      ZeroBasedCounter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "For each interval, each data point is associated with a
        value I, I = 1..N where N is the number of data points;
        reportAggrReportStatSumIX is the multiplication of the
        data point value with the current I.  This value
        along with the other statistics values allow the
        calculation of the slope of the least-squares line
        through the data points."
    ::= { reportStatsDataEntry 12 }

reportStatsDataOverflowStatSumIX  OBJECT-TYPE
    SYNTAX      ZeroBasedCounter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of times the associated
```

```
        reportAggrReportStatSumIX counter has overflowed.
        Note that this object will only be instantiated if the
        associated reportAggrReportHCStatSumIX object is also
        instantiated for a particular dataSource."
    ::= { reportStatsDataEntry 13 }

reportStatsDataHCStatSumIX  OBJECT-TYPE
    SYNTAX      ZeroBasedCounter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The high-capacity version of reportAggrReportStatSumIX.
        Note that this object will only be instantiated if the
        agent supports High Capacity monitoring for a particular
        dataSource."
    ::= { reportStatsDataEntry 14 }

reportStatsDataStatSumIXSq  OBJECT-TYPE
    SYNTAX      ZeroBasedCounter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "For each interval, each data point is associated with a
        value I, I = 1..N where N is the number of data points;
        reportAggrReportStatSumIXSq is the multiplication
        of the data point value squared with the current I.
        This value along with the other statistics
        values allow the calculation of the slope of
        the least-squares line through the data points."
    ::= { reportStatsDataEntry 15 }

reportStatsDataOverflowStatSumIXSq  OBJECT-TYPE
    SYNTAX      ZeroBasedCounter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of times the associated
        reportAggrReportStatSumIXSq counter has overflowed.
        Note that this object will only be instantiated if the
        associated reportAggrReportHCStatSumIXSq object is also
        instantiated for a particular dataSource."
    ::= { reportStatsDataEntry 16 }

reportStatsDataHCStatSumIXSq  OBJECT-TYPE
    SYNTAX      ZeroBasedCounter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
```

```
        "The high-capacity version of reportAggrReportStatSumIXSq.
        Note that this object will only be instantiated if the
        agent supports High Capacity monitoring for a particular
        dataSource."
    ::= { reportStatsDataEntry 17 }

reportStatsDataStatMetricExt1  OBJECT-TYPE
    SYNTAX      ZeroBasedCounter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The .... for the MetricExt1.
        "
    ::= { reportStatsDataEntry 18 }

reportStatsDataStatMetricExt2  OBJECT-TYPE
    SYNTAX      ZeroBasedCounter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The .... for the MetricExt2.
        "
    ::= { reportStatsDataEntry 19 }

reportStatsDataStatMetricExt3  OBJECT-TYPE
    SYNTAX      ZeroBasedCounter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The .... for the MetricExt3.
        "
    ::= { reportStatsDataEntry 20 }

reportStatsDataStatMetricExt4  OBJECT-TYPE
    SYNTAX      ZeroBasedCounter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The .... for the MetricExt4.
        "
    ::= { reportStatsDataEntry 21 }

reportStatsDataStatMetricExt5  OBJECT-TYPE
    SYNTAX      ZeroBasedCounter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The .... for the MetricExt5.
```

```
        "
    ::= { reportStatsDataEntry 22 }




reportSampledGroup       OBJECT IDENTIFIER ::= { reportMIBObjects 2 }

--      Then, the reportSampledGroup assignments are :
--             reportSampledControlTable    - 1
--             reportSampledObjectTable     - 2
--             reportSampledDataTable       - 3


-- REPORT-MIB Editors' Note:
-- The reportSampledGroup is copied from the usrHistory
-- group documented in RMON2 [RFC2021].  We have perserved all of
-- the annotations and object descriptions, as any changes would
-- only diminish the quality of the development.  The only changes
-- made were to the naming of the objects themselves.  Here we have
-- merely prefixed the original names with 'report' and changed the
-- 'usrHistory' to 'Sampled' as we felt this better reflected the
-- the nature of the capability being offered by this group.
-- The remainder of this group development is essentially
-- copied from [RFC2021]:

--
-- Sampled Collection Group (reportSampledGroup)
--
-- The reportSampled group combines mechanisms seen in the alarm and
-- history groups to provide user-specified samplying collection,
-- utilizing two additional control tables and one additional data
-- table. This function has traditionally been done by NMS
-- applications, via periodic polling.  The reportSampled group allows
-- this task to be offloaded to a remote managed device.
--
-- Data (an ASN.1 INTEGER based object) is collected in the same
-- manner as any data table (e.g. etherHistoryTable) except
-- that the user specifies the MIB instances to be collected and their
-- sampling frequency. Objects are collected in
-- bucket-groups, with the intent that all MIB
-- instances in the same bucket-group are collected as atomically as
-- possible by the remote managed device.
--
-- The reportSampledControlTable is a one-dimensional read-create table.
-- Each row configures a collection of sampling buckets; the creation
```

```
-- of a row in this table will cause one or more associated instances in
-- the reportSampledObjectTable to be created. The user specifies the
-- number of bucket elements (rows in the reportSampledObjectTable)
-- requested, as well as the number of buckets requested.
--
-- The reportSampledObjectTable is a 2-d read-write table.
-- Each row configures a single MIB instance to be collected.
-- All rows with the same major index constitute a bucket-group.
--
-- The reportSampledTable is a 3-d read-only table containing
-- the data of associated reportSampledControlEntries. Each
-- entry represents the value of a single MIB instance
-- during a specific sampling interval (or the rate of
-- change during the interval).
--
-- A sample value is stored in two objects - an absolute value and
-- a status object. This allows numbers from -(2G-1) to +4G to be
-- stored.  The status object also indicates whether a sample is
-- valid. This allows data collection to continue if periodic
-- retrieval of a particular instance fails for any reason.
--
-- Row Creation Order Relationships
--
-- The static nature of the reportSampledObjectTable creates
-- some row creation/modification issues. The rows in this
-- table need to be set before the associated
-- reportSampledControlEntry can be activated.
--
-- Note that the reportSampledObject entries associated with a
-- particular reportSampledControlEntry are not required to
-- be active before the control entry is activated. However,
-- the reportSampled data entries associated with an inactive
-- reportSampledObject entry will be inactive (i.e.
-- reportSampledValStatus == valueNotAvailable).
--

reportSampledControlTable OBJECT-TYPE
    SYNTAX SEQUENCE OF SampledControlEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "A list of data-collection configuration entries."
    ::= { reportSampledGroup 1 }


reportSampledControlEntry OBJECT-TYPE
    SYNTAX SampledControlEntry
    MAX-ACCESS not-accessible
```

```
    STATUS current
    DESCRIPTION
        "A list of parameters that set up a group of user-defined
        MIB objects to be sampled periodically (called a
        bucket-group).

        For example, an instance of reportSampledControlInterval
        might be named reportSampledControlInterval.1"
    INDEX { reportSampledControlIndex }
    ::= { reportSampledControlTable 1 }

SampledControlEntry ::= SEQUENCE {
    reportSampledControlIndex             Integer32,
    reportSampledControlObjects           Integer32,
    reportSampledControlBucketsRequested  Integer32,
    reportSampledControlBucketsGranted    Integer32,
    reportSampledControlInterval          Integer32,
    reportSampledControlRequestedNumber   Integer32,
    reportSampledControlReportNumber      Integer32,
    reportSampledControlOwner             OwnerString,
    reportSampledControlStatus            RowStatus
}

reportSampledControlIndex OBJECT-TYPE
    SYNTAX Integer32 (1..65535)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "An index that uniquely identifies an entry in the
        reportSampledControlTable.  Each such entry defines a
        set of samples at a particular interval for a specified
        set of MIB instances available from the managed system."
    ::= { reportSampledControlEntry 1 }

reportSampledControlObjects OBJECT-TYPE
    SYNTAX Integer32 (1..65535)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "The number of MIB objects to be collected
        in the portion of reportSampledTable associated with this
        reportSampledControlEntry.

        This object may not be modified if the associated instance
        of reportSampledControlStatus is equal to active(1)."
    ::= { reportSampledControlEntry 2 }

reportSampledControlBucketsRequested OBJECT-TYPE
```

```
    SYNTAX Integer32 (1..65535)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "The requested number of discrete time intervals
        over which data is to be saved in the part of the
        reportSampledTable associated with this
        reportSampledControlEntry.

        When this object is created or modified, the probe
        should set reportSampledControlBucketsGranted as closely to
        this object as is possible for the particular probe
        implementation and available resources."
    DEFVAL { 50 }
    ::= { reportSampledControlEntry 3 }

reportSampledControlBucketsGranted OBJECT-TYPE
    SYNTAX Integer32 (1..65535)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The number of discrete sampling intervals
        over which data shall be saved in the part of
        the reportSampledTable associated with this
        reportSampledControlEntry.

        When the associated reportSampledControlBucketsRequested
        object is created or modified, the probe should set
        this object as closely to the requested value as is
        possible for the particular probe implementation and
        available resources.  The probe must not lower this
        value except as a result of a modification to the associated
        reportSampledControlBucketsRequested object.

        The associated reportSampledControlBucketsRequested object
        should be set before or at the same time as this object
        to allow the probe to accurately estimate the resources
        required for this reportSampledControlEntry.

        There will be times when the actual number of buckets
        associated with this entry is less than the value of
        this object.  In this case, at the end of each sampling
        interval, a new bucket will be added to the
        reportSampledTable.

        When the number of buckets reaches the value of this object,
        this report is complete and a new report is begun."
    ::= { reportSampledControlEntry 4 }
```

```
reportSampledControlInterval OBJECT-TYPE
    SYNTAX Integer32 (1..2147483647)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "The interval in seconds over which the data is
        sampled for each bucket in the part of the reportSampled
        table associated with this reportSampledControlEntry.

        Because the counters in a bucket may overflow at their
        maximum value with no indication, a prudent manager will
        take into account the possibility of overflow in any of
        the associated counters. It is important to consider the
        minimum time in which any counter could overflow on a
        particular media type and set the
        reportSampledControlInterval object to a value less
        than this interval.

        This object may not be modified if the associated
        reportSampledControlStatus object is equal to active(1)."
    DEFVAL { 1800 }
    ::= { reportSampledControlEntry 5 }

reportSampledControlRequestedNumber OBJECT-TYPE
    SYNTAX Integer32 (1..127)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "The number of reports to be generated and stored by this
         agent for this report request.

        This object may not be modified if the associated
        reportSampledControlStatus object is equal to active(1)."
    DEFVAL { 1 }
    ::= { reportSampledControlEntry 6 }

reportSampledControlReportNumber OBJECT-TYPE
    SYNTAX Integer32 (1..127)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "The number of the current report in progress.  The first
         report is assigned a number equal to '1'.  Each successive
         report number is incremented by unity.  When the last report
         is completed, this value is set to
         reportSampledControlRequestedNumber + 1."
    ::= { reportSampledControlEntry 7 }
```

```
reportSampledControlOwner OBJECT-TYPE
    SYNTAX OwnerString
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "The entity that configured this entry and is
        therefore using the resources assigned to it."
    ::= { reportSampledControlEntry 8 }

reportSampledControlStatus OBJECT-TYPE
    SYNTAX RowStatus
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "The status of this variable history control entry.

        An entry may not exist in the active state unless all
        objects in the entry have an appropriate value.

        If this object is not equal to active(1), all associated
        entries in the reportSampledTable shall be deleted."
    ::= { reportSampledControlEntry 9 }


-- Object table

reportSampledObjectTable OBJECT-TYPE
    SYNTAX SEQUENCE OF SampledObjectEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "A list of data-collection configuration entries."
    ::= { reportSampledGroup 2 }

reportSampledObjectEntry OBJECT-TYPE
    SYNTAX SampledObjectEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "A list of MIB instances to be sampled periodically.

        Entries in this table are created when an associated
        reportSampledControlObjects object is created.

        The reportSampledControlIndex value in the index is
        that of the associated reportSampledControlEntry.

        For example, an instance of reportSampledObjectVariable
```

```
        might be reportSampledObjectVariable.1.3"
    INDEX { reportSampledControlIndex, reportSampledObjectIndex }
    ::= { reportSampledObjectTable 1 }

SampledObjectEntry ::= SEQUENCE {
    reportSampledObjectIndex             Integer32,
    reportSampledObjectVariable          OBJECT IDENTIFIER,
    reportSampledObjectIpAddrType        InetAddressType,
    reportSampledObjectIPAddress         InetAddress,
    reportSampledObjectSampleType        INTEGER
}

reportSampledObjectIndex OBJECT-TYPE
    SYNTAX Integer32 (1..65535)
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An index used to uniquely identify an entry in the
        reportSampledObject table.  Each such entry defines a
        MIB instance to be collected periodically."
    ::= { reportSampledObjectEntry 1 }


reportSampledObjectVariable OBJECT-TYPE
    SYNTAX OBJECT IDENTIFIER
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "The object identifier of the particular variable to be
        sampled.

        Only variables that resolve to an ASN.1 primitive type of
        Integer32 (Integer32, Counter, Gauge, or TimeTicks) may be
        sampled.

        Because SNMP access control is articulated entirely in terms
        of the contents of MIB views, no access control mechanism
        exists that can restrict the value of this object to identify
        only those objects that exist in a particular MIB view.
        Because there is thus no acceptable means of restricting the
        read access that could be obtained through the user history
        mechanism, the probe must only grant write access to this
        object in those views that have read access to all objects on
        the probe.

        During a set operation, if the supplied variable name is not
        available in the selected MIB view, a badValue error must be
        returned.
```

```
        This object may not be modified if the associated
        reportSampledControlStatus object is equal to active(1)."
    ::= { reportSampledObjectEntry 2 }

reportSampledObjectIpAddrType  OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This identifies the IP address type
        of the IP address associated with the
        secondary counter object to be
        monitored within this report.

        This object may not be modified if the associated
        reportStatsControlStatus object is equal to active(1)."
    ::= { reportSampledObjectEntry 3 }

reportSampledObjectIPAddress  OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This identifies the IP addree of the
        secondary counter object to be
        monitored within this report.

        This object may not be modified if the associated
        reportStatsControlStatus object is equal to active(1)."
    ::= { reportSampledObjectEntry 4 }

reportSampledObjectSampleType OBJECT-TYPE
    SYNTAX INTEGER {
              absoluteValue(1),
              deltaValue(2)
          }
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "The method of sampling the selected variable for storage in
        the reportSampledTable.

        If the value of this object is absoluteValue(1), the value of
        the selected variable will be copied directly into the history
        bucket.

        If the value of this object is deltaValue(2), the value of the
        selected variable at the last sample will be subtracted from
```

```
        the current value, and the difference will be stored in the
        history bucket. If the associated reportSampledObjectVariable
        instance could not be obtained at the previous sample
        interval, then a delta sample is not possible, and the value
        of the associated reportSampledValStatus object for this
        interval will be valueNotAvailable(1).

        This object may not be modified if the associated
        reportSampledControlStatus object is equal to active(1)."
    ::= { reportSampledObjectEntry 5 }


-- data table

-- Note: Need to think about how to collect this report data.  It
--   is stored in individual buckets containing individual object
--   samples.  Want to avoid having to table walk to collect this
--   information.
reportSampledTable OBJECT-TYPE
    SYNTAX SEQUENCE OF SampledEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "A list of user defined history entries."
    ::= { reportSampledGroup 3 }

reportSampledEntry OBJECT-TYPE
    SYNTAX SampledEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "A historical sample of user-defined variables.  This sample
        is associated with the reportSampledControlEntry which set
        up the parameters for a regular collection of these samples.

        The reportSampledControlIndex value in the index identifies
        the reportSampledControlEntry on whose behalf this entry
        was created.

        The reportSampledObjectIndex value in the index identifies
        the reportSampledObjectEntry on whose behalf this entry
        was created.

        For example, an instance of reportSampledAbsValue, which
        represents the 14th sample of a variable collected as
        specified by reportSampledControlEntry.1 and
        reportSampledObjectEntry.1.5, would be named
        reportSampledAbsValue.1.14.5"
```

```
    INDEX { reportSampledControlIndex, reportSampledReportIndex,
            reportSampledSampleIndex, reportSampledObjectIndex }
    ::= { reportSampledTable 1 }

SampledEntry ::= SEQUENCE {
    reportSampledReportIndex   Integer32,
    reportSampledSampleIndex   Integer32,
    reportSampledIntervalStart TimeStamp,
    reportSampledIntervalEnd   TimeStamp,
    reportSampledAbsValue      Gauge32,
    reportSampledValStatus     INTEGER
}

reportSampledReportIndex OBJECT-TYPE
    SYNTAX     Integer32 (1..127)
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "An index that uniquely identifies the particular report
        this entry is associated with among the set of reports
        requested through the reportSampledControlNumber in the
        reportSampledControlEntry. This index starts at 1 and
        increases by one as each new report is generated."
    ::= { reportSampledEntry 1 }

reportSampledSampleIndex OBJECT-TYPE
    SYNTAX     Integer32 (1..2147483647)
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "An index that uniquely identifies the particular sample this
        entry represents among all samples associated with the same
        reportSampledControlEntry. This index starts at 1 and
        increases by one as each new sample is taken."
    ::= { reportSampledEntry 2 }

reportSampledIntervalStart OBJECT-TYPE
    SYNTAX TimeStamp
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The value of sysUpTime at the start of the interval over
        which this sample was measured.  If the probe keeps track of
        the time of day, it should start the first sample of the
        history at a time such that when the next hour of the day
        begins, a sample is started at that instant.

        Note that following this rule may require the probe to delay
```

```
        collecting the first sample of the history, as each sample
        must be of the same interval. Also note that the sample which
        is currently being collected is not accessible in this table
        until the end of its interval."
    ::= { reportSampledEntry 3 }

reportSampledIntervalEnd OBJECT-TYPE
    SYNTAX TimeStamp
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The value of sysUpTime at the end of the interval over which
        this sample was measured."
    ::= { reportSampledEntry 4 }

reportSampledAbsValue OBJECT-TYPE
    SYNTAX Gauge32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The absolute value (i.e. unsigned value) of the
        user-specified statistic during the last sampling period. The
        value during the current sampling period is not made available
        until the period is completed.

        To obtain the true value for this sampling interval, the
        associated instance of reportSampledValStatus must
        be checked, and reportSampledAbsValue adjusted as necessary.

        If the MIB instance could not be accessed during the sampling
        interval, then this object will have a value of zero and the
        associated instance of reportSampledValStatus will be set to
        'valueNotAvailable(1)'."
    ::= { reportSampledEntry 5 }


reportSampledValStatus OBJECT-TYPE
    SYNTAX INTEGER {
        valueNotAvailable(1),
        valuePositive(2),
        valueNegative(3)
    }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This object indicates the validity and sign of the data in
        the associated instance of reportSampledAbsValue.
```

```
        If the MIB instance could not be accessed during the sampling
        interval, then 'valueNotAvailable(1)' will be returned.

        If the sample is valid and actual value of the sample is
        greater than or equal to zero then 'valuePositive(2)' is
        returned.

        If the sample is valid and the actual value of the sample is
        less than zero, 'valueNegative(3)' will be returned. The
        associated instance of reportSampledAbsValue should be
        multiplied by -1 to obtain the true sample value."
    ::= { reportSampledEntry 6 }

-- REPORT-MIB Editors' Note:  This ends the copy of definitions from
-- the usrHistory group from RMON2 [RFC 2021].




reportHistoryGroup       OBJECT IDENTIFIER ::= { reportMIBObjects 3 }

--      Then, the reportHistoryGroup assignments are :
--             reportHistoryControlTable    - 1
--             reportHistoryDataTable       - 2

-- Notes: The history group is intended to track changes in
--   identified objects ot type counter, gauge, other.  Each,
--   time the object is updated in the associated MIB, the
--   history group stores a table entry in the associated
--   historyDataTable capturing the time the change was
--   made to the identified object.

--   The historyControl Table ...
--
--   The historyData Table ....

reportHistoryControlTable OBJECT-TYPE
    SYNTAX SEQUENCE OF HistoryControlEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "A list of data-collection configuration entries."
    ::= { reportHistoryGroup 1 }

reportHistoryControlEntry OBJECT-TYPE
    SYNTAX HistoryControlEntry
    MAX-ACCESS not-accessible
```

```
    STATUS current
    DESCRIPTION
        "A list of parameters that set up the collection
        of a history of changes
        in the user-defined MIB objects.

        For example, an instance of reportHistoryControlObject
        might be named reportHistoryControlObject.1"
    INDEX { reportHistoryControlIndex }
    ::= { reportHistoryControlTable 1 }

HistoryControlEntry ::= SEQUENCE {
    reportHistoryControlIndex             Integer32,
    reportHistoryControlObject            OBJECT IDENTIFIER,
    reportHistoryControlObjectIpAddrType  InetAddressType,
    reportHistoryControlObjectIPAddress   InetAddress,
    reportHistoryControlSizeRequested     Integer32,
    reportHistoryControlSizeGranted       Integer32,
    reportHistoryControlRequestedNumber   Integer32,
    reportHistoryControlReportNumber      Integer32,
    reportHistoryControlOwner             OwnerString,
    reportHistoryControlStatus            RowStatus
}

reportHistoryControlIndex OBJECT-TYPE
    SYNTAX Integer32 (1..65535)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "An index that uniquely identifies an entry in the
        reportHistoryControlTable.  Each such entry defines a
        set of histories at a particular interval for a specified
        MIB object instance available from the managed system."
    ::= { reportHistoryControlEntry 1 }

reportHistoryControlObject OBJECT-TYPE
    SYNTAX OBJECT IDENTIFIER
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "The MIB object to be monitored for the collection
        histories in the reportHistoryDataTable associated with this
        reportHistoryControlEntry.

        This object may not be modified if the associated instance
        of reportHistoryControlStatus is equal to active(1)."
    ::= { reportHistoryControlEntry 2 }
```

```
reportHistoryControlObjectIpAddrType  OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This identifies the IP address type
        of the IP address associated with the
        secondary counter object to be
        monitored within this report.

        This object may not be modified if the associated
        reportStatsControlStatus object is equal to active(1)."
    ::= { reportHistoryControlEntry 3 }

reportHistoryControlObjectIPAddress  OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This identifies the IP addree of the
        secondary counter object to be
        monitored within this report.

        This object may not be modified if the associated
        reportStatsControlStatus object is equal to active(1)."
    ::= { reportHistoryControlEntry 4 }

reportHistoryControlSizeRequested OBJECT-TYPE
    SYNTAX Integer32 (1..65535)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "The requested maximum number of history entries
        to be saved in the
        reportHistoryDataTable associated with this
        reportHistoryControlEntry.

        When this object is created or modified, the device
        should set reportHistoryControlSizeGranted as closely to
        this object as is possible for the particular device
        implementation and available resources."
    DEFVAL { 50 }
    ::= { reportHistoryControlEntry 5 }

reportHistoryControlSizeGranted OBJECT-TYPE
    SYNTAX Integer32 (1..65535)
    MAX-ACCESS read-only
    STATUS current
```

```
    DESCRIPTION
        "The maximum allowed number of discrete history entries
        in the reportHistoryTable associated with this
        reportHistoryControlEntry.

        When the associated reportHistoryControlSizeRequested
        object is created or modified, the device should set
        this object as closely to the requested value as is
        possible for the particular device implementation and
        available resources.  The device must not lower this
        value except as a result of a modification to the associated
        reportHistoryControlSizeRequested object.

        The associated reportHistoryControlSizeRequested object
        should be set before or at the same time as this object
        to allow the device to accurately estimate the resources
        required for this reportHistoryControlEntry.

        When the number of histories reaches the value of this object
        and a new history is to be added to the reportHistoryTable,
        the oldest history associated with this
        reportHistoryControlEntry shall be deleted by the agent
        so that the new history can be added."
    ::= { reportHistoryControlEntry 6 }

reportHistoryControlRequestedNumber OBJECT-TYPE
    SYNTAX Integer32 (1..127)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "The number of reports to be generated and stored by this
         agent for this report request.

        This object may not be modified if the associated
        reportHistoryControlStatus object is equal to active(1)."
    DEFVAL { 1 }
    ::= { reportHistoryControlEntry 7 }

reportHistoryControlReportNumber OBJECT-TYPE
    SYNTAX Integer32 (1..127)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "The number of the current report in progress.  The first
         report is assigned a number equal to '1'.  Each successive
         report number is incremented by unity.  When the last report
         is completed, this value is set to
         reportSampledControlRequestedNumber + 1."
```

```
    ::= { reportHistoryControlEntry 8 }

reportHistoryControlOwner OBJECT-TYPE
    SYNTAX OwnerString
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "The entity that configured this entry and is
        therefore using the resources assigned to it."
    ::= { reportHistoryControlEntry 9 }

reportHistoryControlStatus OBJECT-TYPE
    SYNTAX RowStatus
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "The status of this variable history control entry.

        An entry may not exist in the active state unless all
        objects in the entry have an appropriate value.

        If this object is not equal to active(1), all associated
        entries in the reportHistoryTable shall be deleted."
    ::= { reportHistoryControlEntry 10 }



-- data table

-- Note: Similar to the note on the sampled report
--   collection above.  We need to consider what
--   model to use to transmit the report data to
--   the remote management application.  Currently
--   the data is stored in individuals events per
--   table row.  This will impact the design of the
--   table as well as the design of the
--   Notifications.
reportHistoryTable OBJECT-TYPE
    SYNTAX SEQUENCE OF HistoryEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "A list of user defined history entries."
    ::= { reportHistoryGroup 3 }

reportHistoryEntry OBJECT-TYPE
    SYNTAX HistoryEntry
    MAX-ACCESS not-accessible
```

```
    STATUS current
    DESCRIPTION
        "A historical trail of user-defined variables.  This list
        is associated with the reportHistoryControlEntry which set
        up the parameters for a regular collection of these samples.

        The reportHistoryControlIndex value in the index identifies
        the reportHistoryControlEntry on whose behalf this entry
        was created.  This also identifies the MIB object
        being tracked by this reportHistoryEntry.

        For example, an instance of reportHistory...
        "
    INDEX { reportHistoryControlIndex,
            reportHistoryDataIndex }
    ::= { reportHistoryTable 1 }

HistoryEntry ::= SEQUENCE {
    reportHistoryDataIndex        Integer32,
    reportHistoryDataChangeTime   TimeStamp,
    reportHistoryDataValueType    INTEGER,
    reportHistoryDataValue        OCTET STRING,
    reportHistoryDataValStatus    INTEGER
}

reportHistoryDataIndex OBJECT-TYPE
    SYNTAX     Integer32 (1..2147483647)
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "An index that uniquely identifies the particular sample this
        entry represents among all historical entries
        associated with the same
        reportHistoryControlEntry. This index starts at 1 and
        increases by one as each new sample is taken."
    ::= { reportHistoryEntry 1 }

reportHistoryDataChangeTime   OBJECT-TYPE
    SYNTAX TimeStamp
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The value of sysUpTime at the time that the MIB object was
        updated."
    ::= { reportHistoryEntry 2 }

-- Note: May want to move this to the reportHistoryControlTable,
--    as it is too redundant in this table.  Also, need to reconsider
```

```
--    the best way to indicate type and to represent values.
reportHistoryDataValueType OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The type of the data value stored in the
        reportHistoryDataValue string.  The user identifies
        the MIB object to be tracked by this table.
        Various types of objects can be track, so the
        application needs to know the data type being
        stored.  Types supported include counter, gauge,
        integer, float.
        "
    ::= { reportHistoryEntry 3 }

reportHistoryDataValue  OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The absolute value of the
        user-specified MIB object tracked by this
        table entry.  This holds the new object
        value following this change in value.

        If the MIB instance could not be accessed ....
        "
    ::= { reportHistoryEntry 4 }

-- Note: Need to consider in detail the ability of the
--   device to track the times of object change in
--   enough detial to be useful.  What happens if the
--   device gets too busy and delays updating MIB object
--   values tracked by this table entry.  Needs more work.
reportHistoryDataValStatus OBJECT-TYPE
    SYNTAX INTEGER {
        valueAvailable(1),
        valueDelayed(2)
    }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This object indicates the validity of the data in
        the associated instance of reportHistoryAbsValue.

        If the MIB instance could not be accessed promptely,
        then 'valueDelayed(2)' will be returned.
```

```
        If the sample is valid and actual value of the sample
        was promptly recorded, then 'valueAvailable(1)' is
        returned.
        "
    ::= { reportHistoryEntry 5 }



--
-- Notifications
--

-- NOTE: What is the report transmission model we want to
--       support for this MIB?  Want to minimize chatter
--       on the network.  Potentially want to see if
--       can pack reports into Notifications(?).
--       The statsReports are  formatted in a way to
--       support bulk transmissions.  However, as noted
--       above, the sampledReports and the historyReports
--       are stored as individual measurements per row and
--       storage is continually rotaed as more measurements
--       are made in these two report types.  This
--       may complicate report transmission and
--       Notifications definitions.

-- NOTE:  What notifications do we want for this MIB?
--        Checkout what is done in the APM-MIB for Notifications?
--        Examples may include a) report completion
--                             b) overflow counters exceeded

reportNotificationControl OBJECT IDENTIFIER
                                 ::= {reportMIBNotifications 1}
reportNotificationObjects OBJECT IDENTIFIER
                                 ::= {reportMIBNotifications 2}
reportNotificationStates  OBJECT IDENTIFIER
                                 ::= {reportMIBNotifications 3}


   -- reportNotificationControl

   reportSetNotification OBJECT-TYPE
          SYNTAX       OCTET STRING (SIZE(4))
          MAX-ACCESS   read-write
          STATUS       current
          DESCRIPTION
             "A 4-octet string serving as a bit map for
             the notification events defined by the REPORT
             notifications. This object is used to enable
```

```
             and disable specific REPORT notifications where
             a 1 in the bit field represents enabled. The
             right-most bit (least significant) represents
             notification 0.

             This object is persistent and when written
             the entity SHOULD save the change to
             non-volatile storage.
             "
            ::= { reportNotificationControl 1 }


   -- reportNotificationObjects

   reportNewStatsDataReport NOTIFICATION-TYPE
          OBJECTS { reportStatsControlIndex, -- The index of the
                             --   control table for this report
                    reportStatsDataIndex     -- The index of the
                             --   data table for this report
                  }
          STATUS       current
          DESCRIPTION
             "reportNewStatsDataReport is a notification sent
              when a new report is completed from the
              reportStatsControlTable.  The notification carries
              the index from the control table that established
              this report and the index from the data table that
              holds this report."
          ::= { reportNotificationObjects 1 }

   reportNewSampledDataReport NOTIFICATION-TYPE
          OBJECTS { reportSampledControlIndex, -- The index of the
                             --   control table for this report
                    reportSampledReportIndex   -- The index of the
                             --   data table for this report
                  }
          STATUS       current
          DESCRIPTION
             "reportNewSampledDataReport is a notification sent
              when a new report is completed from the
              reportSampledControlTable.  The notification carries
              the index from the control table that established
              this report and the index from the data table that
              holds this report.  Indication of the new report
              is when the reportSampledControlReportNumber
              is incremented."
          ::= { reportNotificationObjects 2 }
```

```
   reportNewHistoryDataReport NOTIFICATION-TYPE
          OBJECTS { reportHistoryControlIndex, -- The index of the
                             --   control table for this report
                    reportHistoryDataIndex   -- The index of the
                             --   data table for this report
                  }
          STATUS       current
          DESCRIPTION
             "reportNewHistoryDataReport is a notification sent
              when a new report is completed from the
              reportHistoryControlTable.  The notification carries
              the index from the control table that established
              this report and the index from the data table that
              holds this report.  Indication of the new report
              is when the reportHistoryControlReportNumber
              is incremented."
          ::= { reportNotificationObjects 3 }


    -- reportNotificationStates
    --   none to define



--
-- Compliance Statements
--

-- [NOTE: Current thoughts on Conformance follow:
--    Mandatory for Stats will include no extensions,
--    or high capacity objects.
--    Hence, the reports will have only the hard-coded statistics.
--    Optional for Stats will be extensions definition table and high
--    capacity objects.
--
--    Mandatory for Sampled will include all.
--
--    Mandatory for History will include all.]


reportCompliances  OBJECT IDENTIFIER ::= { reportMIBConformance 1 }
reportMIBGroups    OBJECT IDENTIFIER ::= { reportMIBConformance 2 }

reportStatsBasicCompliance  MODULE-COMPLIANCE
   STATUS current
   DESCRIPTION "The Stats basic implementation requirements for
                managed network entities that implement
                the REPORT process."
```

```
   MODULE  -- this module
   MANDATORY-GROUPS {reportStatsCapabilitiesBaseObjectsGroup,
                     reportStatsControlBaseObjectsGroup,
                     reportStatsDataBaseObjectsGroup,
                     reportNotificationGroup,
                     reportStatsNotificationGroup }
::= { reportCompliances 1 }

reportStatsHCCompliance MODULE-COMPLIANCE
   STATUS current
   DESCRIPTION "The HC implementation requirements for
               managed network entities that implement
               the REPORT process."
   MODULE  -- this module
   MANDATORY-GROUPS {reportStatsCapabilitiesBaseObjectsGroup,
                     reportStatsControlBaseObjectsGroup,
                     reportStatsDataBaseObjectsGroup,
                     reportNotificationGroup,
                     reportStatsNotificationGroup,
                     reportStatsDataHCObjectsGroup }
::= { reportCompliances 2 }

reportStatsExtendedMetricsCompliance MODULE-COMPLIANCE
   STATUS current
   DESCRIPTION "The extended metrics implementation requirements for
               managed network entities that implement
               the REPORT process."
   MODULE  -- this module
   MANDATORY-GROUPS {reportStatsCapabilitiesBaseObjectsGroup,
                     reportStatsControlBaseObjectsGroup,
                     reportStatsDataBaseObjectsGroup,
                     reportNotificationGroup,
                     reportStatsNotificationGroup,
                     reportStatsExtendedMetricsCapabilitiesObjectsGroup,
                     reportStatsExtendedMetricsControlObjectsGroup,
                     reportStatsExtendedMetricsDataObjectsGroup }
::= { reportCompliances 3 }

reportSampledBasicCompliance  MODULE-COMPLIANCE
   STATUS current
   DESCRIPTION "The Sampled basic implementation requirements for
               managed network entities that implement
               the REPORT process."
   MODULE  -- this module
   MANDATORY-GROUPS {reportSampledControlBaseObjectsGroup,
                     reportSampledObjectIDBaseObjectsGroup,
                     reportSampledDataBaseObjectsGroup,
                     reportNotificationGroup,
```

```
                     reportSampledNotificationGroup }
::= { reportCompliances 4 }


reportHistoryBasicCompliance  MODULE-COMPLIANCE
   STATUS current
   DESCRIPTION "The History basic implementation requirements for
                managed network entities that implement
                the REPORT process."
   MODULE  -- this module
   MANDATORY-GROUPS {reportHistoryControlBaseObjectsGroup,
                     reportHistoryDataBaseObjectsGroup,
                     reportNotificationGroup,
                     reportHistoryNotificationGroup }
::= { reportCompliances 5 }



-- Units of Conformance

reportStatsCapabilitiesBaseObjectsGroup OBJECT-GROUP
   OBJECTS {
            reportClockResolution,
            reportClockMaxSkew,
            reportClockSource
   }
   STATUS  current
   DESCRIPTION
      "Set of REPORT configuration objects implemented
       in this module."
::= { reportMIBGroups 1 }

reportStatsControlBaseObjectsGroup OBJECT-GROUP
   OBJECTS {
            reportStatsControlIndex,
            reportStatsControlInterval,
            reportStatsControlBinInterval,
            reportStatsControlPriObjID,
            reportStatsControlPriObjIpAddrType,
            reportStatsControlPriObjIPAddr,
            reportStatsControlReqReports,
            reportStatsControlGrantedReports,
            reportStatsControlStartTime,
            reportStatsControlReportNumber,
            reportStatsControlInsertsDenied,
            reportStatsControlOwner,
            reportStatsControlStorageType,
            reportStatsControlStatus
```

```
   }
   STATUS  current
   DESCRIPTION
      "Set of REPORT Stats Control base objects implemented
       in this module."
::= { reportMIBGroups 2 }

reportStatsDataBaseObjectsGroup  OBJECT-GROUP
   OBJECTS {
            reportStatsDataIndex,
            reportStatsDataStatN,
            reportStatsDataStatSumX,
            reportStatsDataOverflowStatSumX,
            reportStatsDataStatMaximum,
            reportStatsDataStatMinimum,
            reportStatsDataStatSumSq,
            reportStatsDataOverflowStatSumSq,
            reportStatsDataStatSumIX,
            reportStatsDataOverflowStatSumIX,
            reportStatsDataStatSumIXSq,
            reportStatsDataOverflowStatSumIXSq
   }
   STATUS  current
   DESCRIPTION
      "Set of REPORT state objects implemented
       in this module."
::= { reportMIBGroups 3 }

reportNotificationGroup  OBJECT-GROUP
   OBJECTS {
            reportSetNotification
   }
   STATUS  current
   DESCRIPTION
      "Set of REPORT notifications implemented
       in this module for the Statistics reports."
::= { reportMIBGroups 4 }

reportStatsNotificationGroup  NOTIFICATION-GROUP
   NOTIFICATIONS {
            reportNewStatsDataReport
   }
   STATUS  current
   DESCRIPTION
      "Set of REPORT notifications implemented
       in this module for the Statistics reports."
::= { reportMIBGroups 5 }
```

```
reportStatsDataHCObjectsGroup  OBJECT-GROUP
   OBJECTS {
            reportStatsDataHCStatSumX,
            reportStatsDataHCStatSumSq,
            reportStatsDataHCStatSumIX,
            reportStatsDataHCStatSumIXSq
   }
   STATUS  current
   DESCRIPTION
      "Set of REPORT state objects implemented
       in this module."
::= { reportMIBGroups 6 }

reportStatsExtendedMetricsCapabilitiesObjectsGroup  OBJECT-GROUP
   OBJECTS {
            reportMetricExtDefType,
            reportMetricExtDefName,
            reportMetricExtDefOperation,
            reportMetricExtDefReference,
            reportMetricDirLastChange
   }
   STATUS  current
   DESCRIPTION
      "Set of REPORT state objects implemented
       in this module."
::= { reportMIBGroups 7 }

reportStatsExtendedMetricsControlObjectsGroup  OBJECT-GROUP
   OBJECTS {
            reportStatsControlSecObj1ID,
            reportStatsControlSecObj1IpAddrType,
            reportStatsControlSecObj1IPAddr,
            reportStatsControlSecObj2ID,
            reportStatsControlSecObj2IpAddrType,
            reportStatsControlSecObj2IPAddr,
            reportStatsControlSecObj3ID,
            reportStatsControlSecObj3IpAddrType,
            reportStatsControlSecObj3IPAddr,
            reportStatsControlSecObj4ID,
            reportStatsControlSecObj4IpAddrType,
            reportStatsControlSecObj4IPAddr,
            reportStatsControlSecObj5ID,
            reportStatsControlSecObj5IpAddrType,
            reportStatsControlSecObj5IPAddr,
            reportStatsControlMetricExt1,
            reportStatsControlMetricExt2,
            reportStatsControlMetricExt3,
            reportStatsControlMetricExt4,
```

```
            reportStatsControlMetricExt5
   }
   STATUS  current
   DESCRIPTION
      "Set of REPORT state objects implemented
       in this module."
::= { reportMIBGroups 8 }

reportStatsExtendedMetricsDataObjectsGroup  OBJECT-GROUP
   OBJECTS {
            reportStatsDataStatMetricExt1,
            reportStatsDataStatMetricExt2,
            reportStatsDataStatMetricExt3,
            reportStatsDataStatMetricExt4,
            reportStatsDataStatMetricExt5
   }
   STATUS  current
   DESCRIPTION
      "Set of REPORT state objects implemented
       in this module."
::= { reportMIBGroups 9 }

reportSampledControlBaseObjectsGroup  OBJECT-GROUP
   OBJECTS {
            reportSampledControlIndex,
            reportSampledControlObjects,
            reportSampledControlBucketsRequested,
            reportSampledControlBucketsGranted,
            reportSampledControlInterval,
            reportSampledControlRequestedNumber,
            reportSampledControlReportNumber,
            reportSampledControlOwner,
            reportSampledControlStatus
   }
   STATUS  current
   DESCRIPTION
      "Set of REPORT state objects implemented
       in this module."
::= { reportMIBGroups 10 }

reportSampledObjectIDBaseObjectsGroup  OBJECT-GROUP
   OBJECTS {
            reportSampledObjectVariable,
            reportSampledObjectIpAddrType,
            reportSampledObjectIPAddress,
            reportSampledObjectSampleType
   }
   STATUS  current
```

```
   DESCRIPTION
      "Set of REPORT state objects implemented
       in this module."
::= { reportMIBGroups 11 }

reportSampledDataBaseObjectsGroup  OBJECT-GROUP
   OBJECTS {
            reportSampledReportIndex,
            reportSampledIntervalStart,
            reportSampledIntervalEnd,
            reportSampledAbsValue,
            reportSampledValStatus
   }
   STATUS  current
   DESCRIPTION
      "Set of REPORT state objects implemented
       in this module."
::= { reportMIBGroups 12 }

reportSampledNotificationGroup  NOTIFICATION-GROUP
   NOTIFICATIONS {
            reportNewSampledDataReport
   }
   STATUS  current
   DESCRIPTION
      "Set of REPORT notifications implemented
       in this module for the Sampled reports."
::= { reportMIBGroups 13 }

reportHistoryControlBaseObjectsGroup  OBJECT-GROUP
   OBJECTS {
            reportHistoryControlIndex,
            reportHistoryControlObject,
            reportHistoryControlObjectIpAddrType,
            reportHistoryControlObjectIPAddress,
            reportHistoryControlSizeRequested,
            reportHistoryControlSizeGranted,
            reportHistoryControlRequestedNumber,
            reportHistoryControlReportNumber,
            reportHistoryControlOwner,
            reportHistoryControlStatus
   }
   STATUS  current
   DESCRIPTION
      "Set of REPORT state objects implemented
       in this module."
::= { reportMIBGroups 14 }
```

```
reportHistoryDataBaseObjectsGroup  OBJECT-GROUP
   OBJECTS {
            reportHistoryDataIndex,
            reportHistoryDataChangeTime,
            reportHistoryDataValueType,
            reportHistoryDataValue,
            reportHistoryDataValStatus
   }
   STATUS  current
   DESCRIPTION
      "Set of REPORT state objects implemented
       in this module."
::= { reportMIBGroups 15 }

reportHistoryNotificationGroup  NOTIFICATION-GROUP
   NOTIFICATIONS {
            reportNewHistoryDataReport
   }
   STATUS  current
   DESCRIPTION
      "Set of REPORT notifications implemented
       in this module for the History reports."
::= { reportMIBGroups 16 }


END
```

8. Security Considerations

[TODO] Each specification that defines one or more MIB modules MUST contain a section that discusses security considerations relevant to those modules. This section MUST be patterned after the latest approved template (available at http://www.ops.ietf.org/mib-security.html). Remember that the objective is not to blindly copy text from the template, but rather to think and evaluate the risks/vulnerabilities and then state/ document the result of this evaluation.

[TODO] if you have any read-write and/or read-create objects, please include the following boilerplate paragraph.

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 can have a negative effect on network operations. These are the tables and objects and their sensitivity/vulnerability:

- o [TODO] writable MIB objects that could be especially disruptive if abused MUST be explicitly listed by name and the associated security risks MUST be spelled out; RFC 2669 has a very good example.

- o [TODO] list the writable tables and objects and state why they are sensitive.

[TODO] else if there are no read-write objects in your MIB module, use the following boilerplate paragraph.

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.

[TODO] if you have any sensitive readable objects, please include the following boilerplate paragraph.

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:

- o [TODO] you must explicitly list by name any readable objects that are sensitive or vulnerable and the associated security risks MUST be spelled out (for instance, if they might reveal customer information or violate personal privacy laws such as those of the European Union if exposed to unauthorized parties)

- o [TODO] list the tables and objects and state why they are sensitive.

[TODO] discuss what security the protocol used to carry the information should have. The following three boilerplate paragraphs should not be changed without very good reason. Changes will almost certainly require justification during IESG review.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPSec), even then, there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

It is RECOMMENDED that implementers consider the security features as provided by the SNMPv3 framework (see [RFC3410], section 8), including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

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.

9. IANA Considerations

[TODO] In order to comply with IESG policy as set forth in http://www.ietf.org/ID-Checklist.html, every Internet-Draft that is submitted to the IESG for publication MUST contain an IANA Considerations section. The requirements for this section vary depending what actions are required of the IANA. see RFC4181 section 3.5 for more information on writing an IANA clause for a MIB module document.

[TODO] select an option and provide the necessary details.

Option #1:

```
     The MIB module in this document uses the following IANA-assigned
     OBJECT IDENTIFIER values recorded in the SMI Numbers registry:

     Descriptor        OBJECT IDENTIFIER value
     ----------        -----------------------

     sampleMIB  { mib-2 XXX }
```

Option #2:

Editor's Note (to be removed prior to publication): the IANA is requested to assign a value for "XXX" under the 'mib-2' sub-tree and to record the assignment in the SMI Numbers registry. When the assignment has been made, the RFC Editor is asked to replace "XXX" (here and in the MIB module) with the assigned value and to remove this note.

Note well: prior to official assignment by the IANA, a draft document MUST use placeholders (such as "XXX" above) rather than actual numbers. See RFC4181 Section 4.5 for an example of how this is done in a draft MIB module.

Option #3:

This memo includes no request to IANA.

10. Contributors

This MIB document uses the template authored by D. Harrington which is based on contributions from the MIB Doctors, especially Juergen Schoenwaelder, Dave Perkins, C.M.Heard and Randy Presuhn.

11. Acknowledgements

We would like to thank Bert Wijnen and Andy Bierman for pointing out the existence of the usrHistory group within RMON2 and in answering our numerous questions on the usrHistory group. Further, we wish to thank U. Herberg for his forcing additions to this MIB through his thoughtful consideration of performance monitoring requirements for other MIBs, e.g., NHDP and OLSR MIBs.

12. References

12.1. Normative References

[RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.

[RFC3418] Presuhn, R., "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3418, December 2002.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

12.2. Informative References

[RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

[RFC1757] Waldbusser, S., "Remote Network Monitoring Management Information Base", RFC 1757, February 1995.

[RFC2021] Waldbusser, S., "Remote Network Monitoring Management Information Base Version 2 using SMIv2", RFC 2021, January 1997.

[RFC4150] Dietz, R. and R. Cole, "Transport Performance Metrics MIB", RFC 4150, August 2005.

Appendix A. Change Log

Changes from draft-ietf-manet-report-mib-00 to draft-ietf-manet-report-mib-01 draft.

1. Proposed additions to the statsReports in order to potentially simplify data transmission to management applications.

2. Added some Notification definitions and their relationship to the three reports' structure, i.e., statsReports, sampledReports, and historyReports.

3. In the process of adding notifications for the Sampled and the History reports, decided to restructure the reports from their previously rolling storage model to the fixed interval reporting used all along in the Statistics reporting. This allows the agent to notify the management application that a report has completed and that it is ready to be pulled from the agent storage.

4. Ran MIB through smilint checker and cleaned up all errors and most warnings. A few warnings remain to be addressed.

5. Cleaned up textual material.

Changes from draft-cole-manet-report-mib-02 to draft-ietf-manet-report-mib-00 draft.

1. Major change was the incorporation of the IP address objects associated with all objects of type 'OBJECT IDENTIFIER'. This allows the REPORT-MIB to exist as a proxy report generation capability on a device separate but in close proximity to the device monitoring the referenced object.

2. Cleaned up the up front text, reducing the repetition with the object descriptions in the MIB.

3. Worked on and added sections discussing the relationship to other MIBs.

Changes from draft-cole-manet-report-mib-01 to draft-cole-manet-report-mib-02 draft.

1. Restructured the MIB somewhat to now offer the three reporting capabilities in increasing order of detail: a) statistical reports, b) sampled reports, and c) historical reports.

2. Renamed the usrHistoryGroup and elements to samplingGroup. This is in line with its actual capabilities.

3. Added a new historyGroup which provides a history of change events.

4. Updated the4 Conformance section to reflect the above changes and additions. But did not yet run smilint to check MIB syntax.

Changes from draft-cole-manet-report-mib-00 to draft-cole-manet-report-mib-01 draft.

1. Added (copied) the usrHistory group from RMON2 into the REPORT-MIB.

2. Restructured the MIB to account for the inclusion of the reportSampledGroup.

3. Dropped the reportCurReportsTable as this did not make sense within the context of the REPORT-MIB.

4. Added the Compliance and Conformance material. Defined several Compliance Groups to all for base implementations of the REPORT-MIB for only statistical reports, for only historical reports or for both. Allow for enhanced implementations to address higher capacity issues and extension to metric reporting for statistical reporting.

5. Ran the MIB through the smilint checker and in the process corrected numerous typos, omissions, TEXTUAL CONVENTIONS, IMPORTS, etc.

6. Updated main text to reflect changes.

Appendix B. Open Issues

This section contains the set of open issues related to the development and design of the REPORT-MIB. This section will not be present in the final version of the MIB and will be removed once all the open issues have been resolved.

1. Need to add an index associated with object IDs of interest which are contained within a table, e.g., IfPacketsIn in an InterfaceTable which is indexed by IfIndex. (Note: (RGC)I think adding the IP address associated with the referenced object addresses this issue.)

2. Complete notification group. Need to develop the preferred data report transmission model. This will influence the design of the Notifications. The initial form for the notifications has been laid out in draft-ietf-manet-report-mib-02.

3. Update the text of the document to reflect the final state of the MIB.

4. Identify all objects requiring non-volatile storage in their DESCRIPTION clauses.

5. Complete the security analysis and section.

6. Cleanup all the [TODOs] from the MIB template.

Appendix C.

```
**************************************************************
* Note to the RFC Editor (to be removed prior to publication) *
*                                                             *
* 1) The reference to RFCXXXX within the DESCRIPTION clauses  *
* of the MIB module point to this draft and are to be         *
* assigned by the RFC Editor.                                 *
*                                                             *
* 2) The reference to RFCXXX2 throughout this document point  *
* to the current draft-ietf-manet-report-xx.txt.  This        *
* need to be replaced with the XXX RFC number.                *
*                                                             *
**************************************************************
```

Authors' Addresses

Robert G. Cole
US Army CERDEC
328 Hopkins Road
Aberdeen Proving Ground, Maryland 21005
USA

Phone: +1 410 278 6779
EMail: robert.g.cole@us.army.mil
URI: http://www.cs.jhu.edu/~rgcole/

Joseph Macker
Naval Research Laboratory
Washington, D.C. 20375
USA

EMail: macker@itd.nrl.navy.mil

Al Morton
AT&T Laboratories
Middletown, N.J. 07724
USA

EMail: amorton@att.com

Network Working Group J. Macker, editor
Internet-Draft NRL
Intended status: Experimental SMF Design Team
Expires: September 15, 2011 IETF MANET WG
March 14, 2011

Simplified Multicast Forwarding
draft-ietf-manet-smf-11

Abstract

This document describes a Simplified Multicast Forwarding (SMF) mechanism that provides basic IP multicast forwarding suitable for wireless mesh and mobile ad hoc network (MANET) use. SMF defines techniques for multicast duplicate packet detection (DPD) to be applied in the forwarding process and includes maintenance and checking operations for both IPv4 and IPv6 protocol use. SMF also specifies mechanisms for applying reduced relay sets to achieve more efficient multicast data distribution within a mesh topology versus simple flooding. The document describes interactions with other protocols and multiple deployment approaches. Distributed algorithms for selecting reduced relay sets and related discussion are provided in the Appendices. Basic issues relating to the operation of multicast MANET border routers are discussed but ongoing work remains in this area beyond the scope of this document.

Status of this Memo

Copyright Notice

Table of Contents

1. Requirements Notation

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 [RFC2119].

2. Introduction and Scope

Unicast routing protocol designs for MANET and wireless mesh use often apply distributed algorithms to flood routing control plane messages within an interior wireless routing domain. For example, algorithms specified within [RFC3626] and [RFC3684] provide distributed methods of dynamically electing reduced relay sets that attempt to efficiently flood routing control messages while maintaining a connected set under dynamic topological conditions.

In one sense, Simplified Multicast Forwarding (SMF) extends the efficient flooding concept to the data forwarding plane. Therefore, SMF provides an appropriate multicast forwarding capability for use cases where localized, efficient flooding is considered an effective design approach. The baseline design is intended to provide a basic, best effort multicast forwarding capability that is constrained to operate within an interior MANET or wireless mesh routing domain. An SMF routing domain is an instance of a SMF routing protocol with common policies that is under a single network administration authority. The main design goals of this SMF specification are to adapt efficient relay sets in MANET type environments [RFC2501] and to define the needed IPv4 and IPv6 multicast duplicate packet detection (DPD) mechanisms to support multi-hop, packet forwarding.

2.1. Terminology

The following abbreviations are used throughout this document:

Abbreviation	Definition
MANET	Mobile Ad hoc Network
SMF	Simplified Multicast Forwarding
CF	Classical Flooding
CDS	Connected Dominating Set
MPR	Multi-Point Relay
S-MPR	Source-based MPR
MPR-CDS	MPR-based CDS
E-CDS	Essential CDS
NHDP	Neighborhood Discovery Protocol
SMF-DPD	SMF-Duplicate Packet Detection
I-DPD	Identification-based DPD
H-DPD	Hash-based DPD
HAV	Hash-assist Value
FIB	Forwarding Information Base
TLV	type-length-value encoding
DoS	Denial of Service

3. Design Overview

Figure 1 provides an overview of the logical SMF node architecture, consisting of "Neighborhood Discovery", "Relay Set Selection" and "Forwarding Process" components. Typically, relay set selection (or self-election) occurs based on dynamic input from a neighborhood discovery process. SMF supports the case where neighborhood discovery and/or relay set selection information is obtained from a coexistent process (e.g., a lower layer mechanism or a unicast routing protocol using relay sets). In some algorithm designs, the forwarding decision for a packet can also depend on previous hop or incoming interface information. The asterisks (*) in Figure 1 mark the primitives and relationships needed by relay set algorithms requiring previous-hop packet forwarding knowledge.

```
             _______________               ______________
            |               |             |              |
            | Neighborhood  |             |  Relay Set   |
            |  Discovery    |------------>|  Selection   |
            |   Protocol    |   neighbor  |  Algorithm   |
            |_______________|     info    |______________|
                    \                            /
                     \                          /
             neighbor\                         /forwarding
               info*  \        ____________   /  status
                       \      |            | /
                        `-->  | Forwarding |<--'
                              |  Process   |
         ~~~~~~~~~~~~~~~~~~>  |____________|~~~~~~~~~~~~~~~~~~>
         incoming packet,                   forwarded packets
         interface id*, and
         previous hop*
```

Figure 1: SMF Node Architecture

There are certain IP multicast packets, defined later in this specification, that are "non-forwardable" and these multicast packets will be ignored by the SMF forwarding engine. The SMF forwarding engine MAY also work with policies and management interfaces to allow additional filtering control over which multicast packets are considered for potential SMF forwarding. This interface would allow more refined dynamic forwarding control once such techniques are matured for MANET operation. At present further discussion of dynamic control is left to future work.

Interoperable SMF implementations MUST use a common DPD approach and be able to process the header options defined in this document for IPv6 operation. We define Classical Flooding (CF), as the simplest case of SMF multicast forwarding. With CF, each SMF router forwards each received multicast packet exactly once. In this case, the need for any relay set selection or neighborhood topology information is eliminated at the expense of additional network overhead. In CF mode, the SMF-DPD functionality is still required. While SMF supports a CF mode of operation the use of more efficient relay set modes is RECOMMENDED to reduce contention and congestion caused by unnecessary packet retransmissions [NTSC99].

An efficient, reduced relay set is realized by selecting and maintaining a subset of all possible routers in a MANET routing domain. Known distributed relay set selection algorithms have demonstrated the ability to provide and maintain a dynamic connected set for forwarding multicast IP packets [MDC04]. A few such relay set selection algorithms are described in the Appendices of this document and the basic designs borrow directly from previously documented IETF work. SMF relay set configuration is extensible and additional relay set algorithms beyond those specified here can be accommodated in future work.

Determining and maintaining an optimized set of forwarding nodes generally requires dynamic neighborhood topology information. Neighborhood topology discovery functions MAY be externally provided by a MANET unicast routing protocol or by using the MANET NeighborHood Discovery Protocol (NHDP) [RFC6130] running in concurrence with SMF. Additionally, this specification allows alternative lower layer interfaces (radio router interface) to provide the necessary neighborhood information to aid in supporting more effective relay set election. Fundamentally, an SMF implementation SHOULD provide the ability for multicast forwarding state to be dynamically managed per operating network interface. Some of the relay state maintenance options and interactions are outlined later in Section 7. This document states specific requirements for neighborhood discovery with respect to the forwarding process and the relay set selection algorithms described herein. For determining dynamic relay sets in the absence of other control interfaces, SMF relies on the MANET NHDP specification to assist in IP layer 2-hop neighborhood state discovery and maintenance for relay set election. "SMF_TYPE" and "SMF_NBR_TYPE" Message and Address Block, respectfully, TLV structures (per [RFC5444]) are defined for use with the NHDP protocol. It is RECOMMENDED that all nodes performing SMF operation in conjunction with NHDP, include these TLV types in any NHDP HELLO messages generated. This capability allows for nodes participating in SMF to be explicitly identified along with their respective dynamic relay set algorithm.

4. SMF Applicability

Within dynamic wireless routing topologies, maintaining traditional forwarding trees to support a multicast routing protocol is often not as effective as in wired networks due to the reduced reliability and increased dynamics of mesh topologies [MGL04] [GM99]. A basic packet forwarding service reaching all connected routers running the SMF protocol within a localized routing domain may provide a useful group communication paradigm for various classes of applications. Applications that could take advantage of a simple multicast forwarding service include multimedia streaming, interactive group-based messaging and applications, peer-to-peer middleware multicasting, and multi-hop mobile discovery or registration services. SMF is likely only appropriate for deployment in limited dynamic wireless routing domains so that the flooding process can be contained. The limited SMF routing domains are further defined as administratively scoped multicast forwarding domains in Section 9.2.

Note again that Figure 1 provides a notional architecture for typical SMF-capable nodes. A goal is that simple leaf nodes may also participate in multicast traffic transmission and reception with standard IP network layer semantics (e.g., special or unnecessary encapsulation of IP packets should be avoided in this case). It is important that SMF deployments in localized edge network settings are able to connect and interoperate with existing standard multicast protocols operating within more conventional Internet infrastructures. A multicast border router or proxy mechanism MUST be used when deployed alongside more fixed-infrastructure IP multicast routing such Protocol Independent Multicast (PIM) variants [RFC3973] and [RFC4601]. Present experimental SMF implementations have demonstrated gateway functionality at MANET border routers operating with existing external IP multicast routing protocols [CDHM07],[DHS08],and [DHG09]. SMF may be extended or combined with other mechanisms to provide increased reliability and group specific filtering, but the details for this are not discussed here.

5. SMF Packet Processing and Forwarding

The SMF Packet Processing and Forwarding actions are conducted with the following packet handling activities:

1. Processing of outbound, locally-generated multicast packets.
2. Reception and processing of inbound packets on specific network interfaces.

The purpose of intercepting outbound, locally-generated multicast packets is to apply any added packet marking needed to satisfy the DPD requirements so that proper forwarding may be conducted. Note that for some system configurations the interception of outbound packets for this purpose is not necessary.

Inbound multicast packets are received by the SMF implementation and processed for possible forwarding. This document does not presently support forwarding of directed broadcast addresses [RFC2644]. SMF implementations MUST be capable of forwarding IP multicast packets with destination addresses that are not node-local and link-local for IPv6 as defined in [RFC4291] and that are not within the local network control block as defined by [RFC5771]

This will help support generic multi-hop multicast application needs or to distribute designated multicast traffic ingressing the SMF routing domain via border routers. The multicast addresses to be forwarded should be maintained by an a priori list or a dynamic forwarding information base (FIB) that MAY interact with future MANET dynamic group membership extensions or management functions. There will also be a well-known multicast group for SMF. This multicast group is specified to contain all routers within an SMF routing domain, so that packets transmitted to the multicast address associated with the group will be delivered to all connected routers running SMF. Due the mobile nature of a MANET, routers running SMF may not be topologically connected at particular times. For IPv6, the multicast address is specified to be "site-local". The name of the multicast group is "SL-MANET-ROUTERS". Minimally SMF MUST forward, as instructed by the relay set selection algorithm, unique (non-duplicate) packets received for this well-known group address when the TTL or hop limit value in the IP header is greater than 1. SMF MUST forward all additional global scope addresses specified within the dynamic FIB or configured list as well. In all cases, the following rules MUST be observed for SMF multicast forwarding:

1. IP multicast packets with TTL <= 1 MUST NOT be forwarded.
2. Link local IP multicast packets MUST NOT be forwarded.
3. Incoming IP multicast packets with an IP source address matching one of those of the local SMF router interface(s) MUST NOT be forwarded.
4. Received frames with the MAC source address matching any MAC address of the routers interfaces MUST NOT be forwarded.
5. Received packets for which SMF cannot reasonably ensure temporal DPD uniqueness MUST NOT be forwarded.
6. When packets are forwarded, TTL or hop limit MUST be decremented by one.

Note that rule #3 is important because over some types of wireless interfaces, the originating SMF router may receive re-transmissions of its own packets when they are forwarded by adjacent routers. This rule avoids unnecessary retransmission of locally-generated packets even when other forwarding decision rules would apply.

An additional processing rule also needs to be considered based upon a potential security threat. As discussed further in Section 10, there may be concern in some SMF deployments that malicious nodes may conduct a denial-of-service attack by remotely "previewing" (e.g., via a directional receive antenna) packets that an SMF node would be forwarding and conduct a "pre-play" attack by transmitting the packet before the SMF node would otherwise receive it but with a reduced TTL (or Hop Limit) field value. This form of attack could cause an SMF node to create a DPD entry that would block the proper forwarding of the valid packet (with correct TTL) through the SMF area. A RECOMMENDED approach to prevent this attack, when it is a concern, would be to cache temporal packet TTL values along with the per-packet DPD state (hash value(s) and/or identifier as described in

Section 6). Then, if a subsequent matching (with respect to DPD) packet arrives with a larger TTL value than the packet that was previously forwarded, SMF should forward the new packet and update the TTL value cached with corresponding DPD state to the new, larger TTL value. There may be temporal cases where SMF would unnecessarily forward some duplicate packets using this approach, but those cases are expected to be minimal and acceptable when compared with the potential threat of denied service.

Once these criteria have been met, an SMF implementation MUST make a forwarding decision dependent upon the relay set selection algorithm in use. One of the requirements of SMF is that it be configured to run a particular relay set selection algorithm when launched. If the SMF implementation is using Classical Flooding (CF), the forwarding decision is implicit once DPD uniqueness is determined. Otherwise, a forwarding decision depends upon the current interface-specific relay set state. The descriptions of the relay set selection algorithms in the Appendices to this document specify the respective heuristics for multicast packet forwarding and specific DPD or other processing required to achieve correct SMF behavior in each case. For example, one class of forwarding is based upon relay set election status and the packet's previous hop, while other classes designate the local SMF router as a forwarder for all neighboring nodes.

6. SMF Duplicate Packet Detection

Duplicate packet detection (DPD) is often a requirement in MANET or wireless mesh packet forwarding mechanisms because packets may be transmitted out the same physical interface upon which they arrived and nodes may also receive copies of previously-transmitted packets from other forwarding neighbors. SMF operation requires DPD and implementations MUST provide mechanisms to detect and reduce the likelihood of forwarding duplicate multicast packets using temporal packet identification. It is RECOMMENDED this be implemented by keeping a history of recently-processed multicast packets for comparison to incoming packets. A DPD packet cache history SHOULD be kept long enough to span the maximum network traversal lifetime, MAX_PACKET_LIFETIME, of multicast packets being forwarded within an SMF routing domain. The DPD mechanism SHOULD avoid keeping unnecessary state for packet flows such as those that are locally-generated or link-local destinations that would not be considered for forwarding as presented in Section 5. For both IPv4 and IPv6, this document describes two basic multicast duplicate packet detection mechanisms: header content identification-based (I-DPD) and hash-based (H-DPD) duplicate detection. I-DPD is a mechanism using specific packet headers, and option headers in the case of IPv6, in combination with flow state to estimate the temporal uniqueness of a packet. H-DPD uses hashing of the particular packet fields and payloads to provide an estimation of temporal uniqueness.

Trade-offs of the two approaches to DPD merit different consideration dependent upon the specific SMF deployment scenario. Because of the potential addition of a hop-by-hop option header with IPv6, SMF deployments MUST be configured to use a common mechanism and DPD algorithm. The main difference between IPv4 and IPv6 SMF-DPD specification is the avoidance of any additional header options in the IPv4 case.

For each network interface, SMF implementations MUST maintain DPD packet state as needed to support the forwarding heuristics of the relay set algorithm used. In general this involves keeping track of previously forwarded packets so that duplicates are not forwarded, but some relay techniques have additional considerations, such as discussed in Appendix B.2.

Additional details of I-DPD and H-DPD processing and maintenance for different classes of packets are described in the following sections.

6.1. IPv6 Duplicate Packet Detection

This section describes the mechanisms and options for SMF IPv6 DPD. The core IPv6 packet header does not provide any explicit identification header field that can be exploited for I-DPD. The following areas are described to support IPv6 DPD and each is covered in more detail in particular subsections:

1. the hop-by-hop SMF-DPD option header,
2. the use of IPv6 fragment header fields for I-DPD when they exist,
3. the use of IPsec sequencing for I-DPD when a non-fragmented, IPsec header is detected, and
4. an H-DPD approach assisted, as needed, by the SMF-DPD option header.

SMF MUST provide a DPD marking module that can insert the hop-by-hop IPv6 header option defined in this section. This process MUST come after any source-based fragmentation that may occur with IPv6. As with IPv4, SMF IPv6 DPD is presently specified to allow either a packet hash or header identification method for DPD. An SMF implementation MUST be configured to operate either in H-DPD or I-DPD mode and perform the appropriate routines outlined in the following sections.

6.1.1. IPv6 SMF-DPD Header Option

The base IPv6 packet header does not contain a unique identifier suitable for DPD. This section defines an IPv6 Hop-by-Hop Option

[RFC2460] to serve this purpose for IPv6 I-DPD. Additionally, the header option provides a mechanism to guarantee non-collision of hash values for different packets when H-DPD is used.

If this is the only hop-by-hop option present, the optional "TaggerId" field (see below) is not included, and the size of the DPD packet identifier (sequence number) or hash token is 24 bits or less, this will result in the addition of 8 bytes to the IPv6 packet header including the "Next Header", "Header Extension Length", SMF-DPD option fields, and padding.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                  ...              |0|0|0| OptType | Opt. Data Len |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |H|  DPD Identifier Option Fields or Hash Assist Value  ...
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Fig. 2 - IPv6 SMF-DPD Hop-by-Hop Header Option

"Option Type" = (Lower 5 bits pending IANA assignment, highest order MUST be 000). By having these three bits be zero, this specification requires that nodes not recognizing this option type should skip over this option and continue processing the header and that the option must not change en route [RFC2460].

"Opt. Data Len" = Length of option content (I.e., 1 + (<IdType> ? (<IdLen> + 1): 0) + Length(DPD ID)).

"H-bit" = a hash indicator bit value identifying DPD marking type. 0 == sequence-based approach w/ optional taggerId and a tuple-based sequence number. 1 == indicates a hash assist value (HAV) field follows to aid in avoiding hash-based DPD collisions.

When the "H-bit" is cleared (zero value), the SMF-DPD format to support I-DPD operation is specified as shown in Figure 2 and defines the extension header in accordance with [RFC2460].

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                ...              |0|0|0|  OptType  | Opt. Data Len |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |0|TidTyp|TidLen|             TaggerId (optional) ...            |
 +-+-+-+-+-+-+-+-+               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                               |            Identifier  ...
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: IPv6 SMF-DPD Header Option in I-DPD mode

The "TidType" is a 3-bit field indicating the presence and type of the optional "TaggerId" field. The optional "TaggerId" is used to differentiate multiple ingressing border gateways that may commonly apply the SMF-DPD option header to packets from a particular source. This is provided for experimental purposes. The following table lists the valid TaggerId types:

Name	Value	Purpose
NULL	0	Indicates no "TaggerId" field is present. "TidLen" MUST also be set to ZERO.
DEFAULT	1	A "TaggerId" of non-specific context is present. "TidLen + 1" defines the length of the TaggerId field in bytes.
IPv4	2	A "TaggerId" representing an IPv4 address is present. The "TidLen" MUST be set to 3.
IPv6	3	A "TaggerId" representing an IPv6 address is present. The "TidLen" MUST be set to 15.
ExtId	7	RESERVED FOR FUTURE USE (possible extended ID)

Table 1: TaggerId Types

This format allows a quick check of the "TidType" field to determine if a "TaggerId" field is present. If the <TidType> is NULL, then the length of the DPD packet <Identifier> field corresponds to the (<Opt. Data Len> - 1). If the <TidType> is non-NULL, then the length of the "TaggerId" field is equal to (<TidLen> - 1) and the remainder of the option data comprises the DPD packet <Identifier> field. When the "TaggerId" field is present, the <Identifier> field can be considered a unique packet identifier in the context of the <taggerId:srcAddr: dstAddr> tuple. When the "TaggerId" field is not present, then it is assumed the source host applied the SMF-DPD option and the <Identifier> can be considered unique in the context of the IPv6 packet header <srcAddr:dstAddr> tuple. IPv6 I-DPD operation details are described in Section 6.1.2.

When the "H-bit" in the SMF-DPD option data is set, the data content value is interpreted as a Hash-Assist Value (HAV) used to facilitate H-DPD operation. In this case, source hosts or ingressing gateways apply the SMF-DPD with a HAV only when required to differentiate the hash value of a new packet with respect to hash values in the DPD cache. This situation can be detected locally on the node by running the hash algorithm and checking the DPD cache. prior ingressing a previously unmarked packet or a locally sourced packet. This helps to guarantee the uniqueness of generated hash values when H-DPD is used. Additionally, this also avoids the added overhead of applying the SMF-DPD option header to every packet. For many hash algorithms, it is expected that only sparse use of the SMF-DPD option may be required. The format of the SMF-DPD header option for H-DPD operation is given in Figure 3.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
              ...              |0|0|0|  OptType    | Opt. Data Len |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1|    Hash Assist Value (HAV) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: IPv6 SMF_DPD Header Option in H-DPD Mode

The SMF-DPD option should be applied with a HAV to produce a unique hash digest for packets within the context of the IPv6 packet header <srcAddr>. The size of the HAV field is implied by the "Opt. Data Len". The appropriate size of the field depends upon the collision properties of the specific hash algorithm used. More details on IPv6 H-DPD operation are provided in Section 6.1.3.

6.1.2. IPv6 Identification-based DPD

The following table summarizes the IPv6 I-DPD processing and forwarding decision approach. Within the table '*' indicates an ignore field condition.

IPv6 Fragment Header	IPv6 IPsec Header	IPv6 I-DPD Header	SMF IPv6 I-DPD Mode Action
Present	*	*	Use Fragment Header I-DPD Check and Process for Forwarding
Not Present	Present	*	Use IPsec Header I-DPD Check and Process for Forwarding
Present	*	Present	Invalid, do not Forward
Not Present	Present	Present	Invalid, do not Forward
Not Present	Not Present	Not Present	Add I-DPD Header,and Process for Forwarding
Not Present	Not Present	Present	Use I-DPD Header Check and Process for Forwarding

Table 2: IPv6 I-DPD Processing Rules

If the IPv6 multicast packet is an IPv6 fragment, SMF MUST use the fragment extension header fields for packet identification. This identifier can be considered unique in the context of the <srcAddr: dstAddr> of the IP packet. If the packet is an unfragmented IPv6 IPsec packet, SMF MUST use IPsec fields for packet identification. The IPsec header <sequence> field can be considered a unique identifier in the context of the <IPsecType:srcAddr:dstAddr:SPI> where the "IPsecType" is either AH or ESP [RFC4302]. For unfragmented, non-IPsec, IPv6 packets, the use of the SMF-DPD header option is necessary to support I-DPD operation. The SMF-DPD header option is applied in the context of the <srcAddr> of the IP packet. End systems or ingressing SMF gateways are responsible for applying this option to support DPD. The following table summarizes these packet identification types:

IPv6 Packet Type	Packet DPD ID Context	Packet DPD ID
Fragment	<srcAddr:dstAddr>	<fragmentOffset:id>
IPsec Packet	<IPsecType:srcAddr:dstAddr:SPI>	<sequence>
Regular Packet	<[taggerId:]srcAddr:dstAddr>	<SMF-DPD option header id>

Table 3: IPv6 I-DPD Packet Identification Types

"IPsecType" is either Authentication Header (AH) or Encapsulating Security Payload (ESP).

The "TaggerId" is an optional field of the IPv6 SMF-DPD header option.

6.1.3. IPv6 Hash-based DPD

A default hash-based DPD approach (H-DPD) for use by SMF is specified as follows. An MD5 [RFC1321] hash of the non-mutable header fields, options fields, and data content of the IPv6 multicast packet is used to produce a 128-bit digest. The least significant 64 bits of this digest is used for SMF packet identification. The approach for calculating this hash value SHOULD follow the same guidelines described for calculating the Integrity Check Value (ICV) described in [RFC4302] with respect to non-mutable fields. This approach should have a reasonably low probability of digest collision when packet headers and content are varying. MD5 is being applied in SMF only to provide a low probability of collision and is not being used for cryptographic or authentication purposes. A history of the packet hash values SHOULD be maintained within the context of the IPv6 packet header <srcAddr>. SMF ingress points (i.e., source hosts or gateways) use this history to confirm that new packets are unique with respect to their hash value. The Hash-assist Value (HAV) field described in Section 6.1.1 is provided as a differentiating field when a digest collision would otherwise occur. Note that the HAV is an immutable option field and SMF MUST process any included HAV values (see Section 6.1.1) in its hash calculation.

If a packet results in a digest collision (i.e., by checking the H-DPD digest history) within the DPD cache kept by SMF forwarders, the packet should be silently dropped. If a digest collision is detected at an SMF ingress point the H-DPD option header is constructed with a randomly generated HAV. A HAV is recalculated as needed to produce a non-colliding hash value prior to forwarding. The multicast packet is then forwarded with the added IPv6 SMF-DPD header option.

The MD5 indexing and IPv6 HAV approaches are specified at present for consistency and robustness to suit experimental uses. Future approaches and experimentation may discover designs tradeoffs in hash robustness and efficiency worth considering. Enhancements MAY include reducing the maximum payload length that is processed, determining shorter indexes, or applying more efficient hashing algorithms. Use of the HAV functionality may allow for application of "lighter-weight" hashing techniques that might not have been initially considered due to poor collision properties otherwise. Such techniques could reduce packet processing overhead and memory requirements.

6.2. IPv4 Duplicate Packet Detection

This section describes the mechanisms and options for IPv4 DPD. The IPv4 packet header [RFC0791] 16-bit "Identification" field MAY be used for DPD assistance, but practical limitations may require alternative approaches in some situations. The following areas are described to support IPv4 DPD:

1. the use of IPv4 fragment header fields for I-DPD when they exist,
2. the use of IPsec sequencing for I-DPD when a non-fragmented IPv4 IPsec packet is detected, and
3. a H-DPD approach.

A specific SMF-DPD marking option is not specified for IPv4 since header options are not as tractable for end systems as for IPv6. IPv4 packets from a particular source are assumed to be marked with a temporally unique value in the "Identification" field of the packet header that can serve for SMF-DPD purposes. However, in present operating system networking kernels, the IPv4 header "Identification" value is not always generated properly, especially when the "don't fragment" (DF) bit is set. The IPv4 I-DPD mode of this specification requires that IPv4 "Identification" fields are managed reasonably by source hosts and that temporally unique values are set within the context of the packet header <protocol:srcAddr:dstAddr> tuple. If this is not expected during an SMF deployment, then it is RECOMMENDED that the H-DPD method be used as a more reliable approach.

Since IPv4 SMF does not specify an options header, the interoperability constraints are looser than the IPv6 version and forwarders may be operate with mixed H-DPD and I-DPD modes as long as they consistently perform the appropriate DPD routines outlined in the following sections. However, it is RECOMMENDED that a deployment be configured with a common mode for operational consistency.

6.2.1. IPv4 Identification-based DPD

The following table summarizes the IPv4 I-DPD processing approach once a packet has passed the basic forwardable criteria described in Section 5. Within the table '*' indicates an ignore field condition. DF, MF, Fragment offset correspond to related fields and flags defined in [RFC0791].

DF flag	MF flag	Fragment offset	IPsec	IPv4 I-DPD Action
1	1	*	*	Invalid, Do Not Forward
1	0	nonzero	*	Invalid, Do Not Forward
*	0	zero	not Present	Tuple I-DPD Check and Process for Forwarding
*	0	zero	Present	IPsec enhanced Tuple I-DPD Check and Process for Forwarding
0	0	nonzero	*	Extended Fragment Offset Tuple I-DPD Check and Process for Forwarding
0	1	zero or nonzero	*	Extended Fragment Offset Tuple I-DPD Check and Process for Forwarding

Table 4: IPv4 I-DPD Processing Rules

For performance reasons, IPv4 network fragmentation and reassembly of multicast packets within wireless MANET networks should be minimized, yet SMF provides the forwarding of fragments when they occur. If the IPv4 multicast packet is a fragment, SMF MUST use the fragmentation header fields for packet identification. This identification can be considered temporally unique in the context of the <protocol:srcAddr: dstAddr> of the IPv4 packet. If the packet is an unfragmented IPv4 IPsec packet, SMF MUST use IPsec fields for packet identification. The IPsec header <sequence> field can be considered a unique identifier in the context of the <IPsecType:srcAddr:dstAddr:SPI> where the "IPsecType" is either AH or ESP [RFC4302]. Finally, for unfragmented, non-IPsec, IPv4 packets, the "Identification" field can be used for I-DPD purposes. The "Identification" field can be considered unique in the context of the IPv4 <protocol:scrAddr: dstAddr> tuple. The following table summarizes these packet identification types:

IPv4 Packet Type	Packet Identification Context	Packet Identifier
Fragment	<protocol:srcAddr:dstAddr>	<fragmentOffset:id>
IPsec Packet	<IPsecType:srcAddr:dstAddr:SPI>	<sequence>

Regular Packet	<protocol:srcAddr:dstAddr>	<identification field>

Table 5: IPv4 I-DPD Packet Identification Types

"IPsecType" is either Authentication Header (AH) or Encapsulating Security Payload (ESP).

The limited size (16 bits) of the IPv4 header "Identification" field [RFC0791] may result in more frequent value field wrapping, particularly if a common sequence space is used by a source for multiple destinations. If I-DPD operation is required, the use of the "internal hashing" technique described in Section 10 may mitigate this limitation of the IPv4 "Identification" field for SMF-DPD. In this case the "internal hash" value would be concatenated with the "Identification" value for I-DPD operation.

6.2.2. IPv4 Hash-based DPD

To ensure consistent IPv4 H-DPD operation among SMF nodes, a default hashing approach is specified. This is similar to that specified for IPv6, but the H-DPD header option with HAV is not considered. SMF MUST perform an MD5 [RFC1321] hash of the immutable header fields, option fields and data content of the IPv4 multicast packet resulting in a 128-bit digest. The least significant 64 bits of this digest is used for SMF packet identification. The approach for calculating the hash value SHOULD follow the same guidelines described for calculating the Integrity Check Value (ICV) described in [RFC4302] with respect to non-mutable fields. A history of the packet hash values SHOULD be maintained in the context of <protocol:srcAddr: dstAddr>. The context for IPv4 is more specific than that of IPv6 since the SMF-DPD HAV cannot be employed to mitigate hash collisions.

The MD5 hash is specified at present for consistency and robustness. Future approaches and experimentation may discover design tradeoffs in hash robustness and efficiency worth considering for future revisions of SMF. This MAY include reducing the packet payload length that is processed, determining shorter indexes, or applying a more efficient hashing algorithm.

7. Relay Set Selection

7.1. Non-Reduced Relay Set Forwarding

SMF implementations MUST support CF as a basic forwarding mechanism when reduced relay set information is not available or not selected for operation. In CF mode, each node transmits a locally generated or newly received forwardable packet exactly once. The DPD techniques described in Section 6 are critical to proper operation and prevent duplicate packet retransmissions by the same forwarding node.

7.2. Reduced Relay Set Forwarding

MANET reduced relay sets are often achieved by distributed algorithms that can dynamically calculate a topological connected dominating set (CDS).

A goal of SMF is to apply reduced relay sets for more efficient multicast dissemination within dynamic topologies. To accomplish this SMF MUST support the ability to modify its multicast packet forwarding rules based upon relay set state received dynamically during operation. In this way, SMF forwarding operates effectively as neighbor adjacencies or multicast forwarding policies within the topology change.

In early SMF experimental prototyping, the relay set information has been derived from coexistent unicast routing control plane traffic flooding processes [MDC04]. From this experience, extra pruning considerations were sometimes required when utilizing a relay set from a separate routing protocol process. As an example, relay sets formed for the unicast control plane flooding MAY include additional redundancy that may not be desired for multicast forwarding use (e.g., biconnected relay set).

Here is a recommended criteria list for SMF relay set selection algorithm candidates:

1. Robustness to topological dynamics and mobility
2. Localized election or coordination of any relay sets
3. Reasonable minimization of CDS relay set size given above constraints
4. Heuristic support for preference or election metrics

Some relay set algorithms meeting these criteria are described in the Appendices of this document. Additional relay set selection algorithms may be specified in separate specifications in the future. Each Appendix subsection in this document can serve as a template for specifying additional relay algorithms.

Figure 4 depicts a information flow diagram of possible relay set control options. The SMF Relay Set State represents the information base that is used by SMF in the forwarding decision process. The relay set control option diagram demonstrates that the SMF relay set state may be determined by fundamentally three different methods: independent operation with NHDP [RFC5444] input providing dynamic network neighborhood adjacency information that is then used by a particular relay set selection, slave operation with an existing unicast MANET routing protocol that is capable of providing CDS election information that can be used by SMF, and cross layer operation that may involve lower layer neighbor or link information. Other heuristics to influence and control election can come from network management or other interfaces as shown on the right. Of course CF mode, simplifies the control and does not require other input but relies solely on DPD.

```
                       Possible L2 Trigger/Information
                                      |
                                      |
    ______________              ______v_____        __________________
   |    MANET      |           |            |       |                  |
   | Neighborhood  |           | Relay Set  |       | Other Heuristics |
   |  Discovery    |----------->| Selection  |<------| (Preference,etc) |
   |   Protocol    | neighbor  | Algorithm  |       |  Net Management  |
   |______________|   info    |____________|       |__________________|
          \                             /
           \                           /
    neighbor\                         / Dynamic Relay
      info*  \   ____________        /    Set Status
              \ |    SMF     |      /  (State, {neighbor info})
               `-->| Relay Set  |<--'
                   |   State    |
                -->|____________|
               /
              /
    ______________
   |  Coexistent   |
   |    MANET      |
   |   Unicast     |
   |   Process     |
   |______________|
```

Figure 4: SMF Reduced Relay Set Information Flow

More discussion is provided on the three styles of SMF operation with reduced relay sets as illustrated in Figure 4 :

1. Independent operation: In this case, SMF operates independently from any unicast routing protocols. To support reduced relay sets SMF MUST perform its own relay set selection using information gathered from signaling. It is RECOMMENDED that an associated MANET NHDP process be use for this signaling. NHDP

messaging SHOULD be appended with additional [RFC5444] type-length-value (TLV) content to support SMF-specific requirements as discussed in [RFC6130] and for the applicable relay set algorithm described in the Appendices of this document or future specifications. Unicast routing protocols may co-exist, even using the same NHDP process, but signaling that supports reduced relay set selection for SMF is independent of these protocols.

2. Operation with CDS-aware unicast routing protocol: In this case, a coexistent unicast routing protocol provides dynamic relay set state based upon its own control plane CDS or neighborhood discovery information.
3. Cross-layer Operation: In this case, SMF operates using neighborhood status and triggers from a cross-layer information base for dynamic relay set selection and maintenance (e.g., lower link layer).

8. SMF Neighborhood Discovery Requirements

This section defines the requirements for use of the MANET Neighborhood Discovery Protocol (NHDP) [RFC6130] to support SMF operation. Note that basic CF forwarding requires no neighborhood topology knowledge since in this configured mode every SMF node relays all traffic. Supporting more reduced SMF relay set operation requires the discovery and maintenance of dynamic neighborhood topology information. The MANET NHDP protocol can be leveraged provide this necessary information, however there are SMF-specific requirements for related NHDP use. This is the case for both "independent" SMF operation where NHDP is being used specifically to support SMF or when one NHDP instance is used for both for SMF and a coexistent MANET unicast routing protocol.

NHDP HELLO messages and the resultant neighborhood information base are described separately within the NHDP specification. To summarize, the NHDP protocol provides the following basic functions:

1. 1-hop neighbor link sensing and bidirectionality checks of neighbor links,
2. 2-hop neighborhood discovery including collection of 2-hop neighbors and connectivity information,
3. Collection and maintenance of the above information across multiple interfaces, and
4. A method for signaling SMF information throughout the 2-hop neighborhood through the use of TLV extensions.

Appendices (A-C) of this document describe CDS-based relay set selection algorithms that can achieve efficient SMF operation, even in dynamic, mobile networks and each of the algorithms has been initially experimented within a working SMF prototype [MDDA07]. When using these algorithms in conjunction with NHDP, a method verifying neighbor SMF operation is required in order to insure correct relay set selection. NHDP along with SMF operation verification provides the necessary information required by these algorithms to conduct relay set selection. Verification of SMF operation may be done administratively or through the use of the SMF relay algorithms TLVs defined in the following subsections. Use of the SMF relay algorithm TLVs is RECOMMENDED when using NHDP for SMF neighborhood discovery.

The following sub-sections specify some SMF-specific TLV types supporting general SMF operation or supporting the algorithms described in the Appendices. The Appendices describing several relay set algorithms also specify any additional requirements for use with NHDP and reference the applicable TLV types as needed.

8.1. SMF Relay Algorithm TLV Types

This section specifies TLV types to be used within NHDP messages to identify the CDS relay set selection algorithm(s) in use. Two TLV types are defined, one message TLV type and one address TLV type.

8.1.1. SMF Message TLV Type

The message TLV type denoted SMF_TYPE is used to identify the existence of an SMF instance operating in conjunction with NHDP. This message TLV type makes use of the extended type field as defined by [RFC5444] to convey the CDS relay set selection algorithm currently in use by the SMF message originator. When NHDP is used to support SMF operation, the SMF_TYPE TLV, containing the extended type field with the appropriate value, SHOULD be included in NHDP_HELLO messages (HELLO messages as defined in [RFC6130]. This allows SMF nodes to learn when neighbors are configured to use NHDP for information exchange including algorithm type and related algorithm information. This information can be used to take action, such as ignoring neighbor information using incompatible algorithms. It is possible that SMF neighbors MAY be configured differently and still operate cooperatively, but these cases will vary dependent upon the algorithm types designated.

This document defines the following Message TLV type as specified in Table 6 conforming to [RFC5444]. The TLV extended type field is used to contain the sender's "Relay Algorithm Type". The interpretation of the "value" content of these TLVs is defined per "Relay Algorithm Type" and may contain algorithm specific information.

	TLV syntax	Field Values
type	<tlv-type>	SMF_TYPE
extended type	<tlv-type-ext>	<relayAlgorithmId>
length	<length>	variable
value	<value>	variable

Table 6: SMF Type Message TLV

In Table 6 <relayAlgorithmId> is an 8-bit field containing a number 0-255 representing the "Relay Algorithm Type" of the originator address of the corresponding NHDP message.

Possible values for the <relayAlgorithmId> are defined in Table 7. The table provides value assignments, future IANA assignment spaces, and an experimental space. The experimental space use MUST NOT assume uniqueness and thus should not be used for general interoperable deployment prior to official IANA assignment.

Type Value	Extended Type Value	Algorithm
SMF_TYPE	0	CF
SMF_TYPE	1	S-MPR
SMF_TYPE	2	E-CDS
SMF_TYPE	3	MPR-CDS
SMF_TYPE	4-127	Future Assignment STD action
SMF_TYPE	128-239	No STD action required
SMF_TYPE	240-255	Experimental Space

Table 7: SMF Relay Algorithm Type Values

Acceptable <length> and <value> fields of an SMF_TYPE TLV are dependent on the extended type value (i.e. relay algorithm type). The appropriate algorithm type, as conveyed in the <tlv-type-ext> field, defines the meaning and format of its TLV <value> field. For the algorithms defined by this document, see the appropriate appendix for the <value> field format.

8.1.2. SMF Address Block TLV Type

An address block TLV type, denoted SMF_NBR_TYPE (i.e., SMF neighbor relay algorithm) is specified in Table 8. This TLV enables CDS relay algorithm operation and configuration to be shared among 2-hop neighborhoods. Some relay algorithms require two hop neighbor configuration in order to correctly select relay sets. It is also useful when mixed relay algorithm operation is possible, some examples of mixed use is outlined in the appendices.

The message SMF_TYPE TLV and address block SMF_NBR_TYPE TLV types share a common format.

	TLV syntax	Field Values
type	<tlv-type>	SMF_NBR_TYPE
extended type	<tlv-type-ext>	<relayAlgorithmId>
length	<length>	variable
value	<value>	variable

Table 8: SMF Type Address Block TLV

<relayAlgorithmId> in Table 8 is an 8-bit unsigned integer field containing a number 0-255 representing the "Relay Algorithm Type" value that corresponds to any associated address in the address block. Note that "Relay Algorithm Type" values for 2-hop neighbors can be conveyed in a single TLV or multiple value TLVs as described in [RFC5444]. It is expected that SMF nodes using NHDP construct address blocks with SMF_NBR_TYPE TLVs to advertise "Relay Algorithm Type" and to advertise neighbor algorithm values received in SMF_TYPE TLVs from those neighbors.

Again values for the <relayAlgorithmId> are defined in Table 8.

The interpretation of the "value" field of SMF_NBR_TYPE TLVs is defined per "Relay Algorithm Type" and may contain algorithm specific information. See the appropriate appendix for definitions of value fields for the algorithms defined by this document.

9. SMF Border Gateway Considerations

It is expected that SMF will be used to provide simple forwarding of multicast traffic within a MANET or mesh routing topology. A border router gateway approach should be used to allow interconnection of SMF areas with networks using other multicast routing protocols, such as PIM. It is important to note that there are many scenario-specific issues that should be addressed when discussing border multicast routers. At the present time, experimental deployments of SMF and PIM border router approaches have been demonstrated[DHS08]. Some of the functionality border routers may need to address includes the following:

1. Determining which multicast group traffic transits the border router whether entering or exiting the attached SMF routing domain.
2. Enforcement of TTL threshold or other scoping policies.
3. Any marking or labeling to enable DPD on ingressing packets.
4. Interface with exterior multicast routing protocols.
5. Possible operation with multiple border routers (presently beyond scope of this document).
6. Provisions for participating non-SMF nodes.

Each of these areas is discussed in more detail in the following subsections. Note the behavior of SMF border routers is the same as that of non-border SMF nodes when forwarding packets on interfaces within the SMF routing domain. Packets that are passed outbound to interfaces operating fixed-infrastructure multicast routing protocols SHOULD be evaluated for duplicate packet status since present standard multicast forwarding mechanisms do not usually perform this function.

9.1. Forwarded Multicast Groups

Mechanisms for dynamically determining groups for forwarding into a MANET SMF routing domain is an evolving technology area. Ideally, only groups for which there is active group membership should be injected into the SMF domain. This can be accomplished by providing an IPv4 Internet Group Membership Protocol (IGMP) or IPv6 Multicast Listener Discovery (MLD) proxy protocol so that MANET SMF nodes can inform attached border routers (and hence multicast networks) of their current group membership status. For specific systems and services it may be possible to statically configure group membership joins in border routers, but it is RECOMMENDED that some form of IGMP/MLD proxy or other explicit, dynamic control of membership be provided. Specification of such an IGMP/MLD proxy protocol is beyond the scope of this document.

For outbound traffic, SMF border routers can perform duplicate packet detection and forward non-duplicate traffic that meets TTL/hop limit and scoping criteria and forward packet to interfaces external to the SMF routing domain. Appropriate IP multicast routing (PIM, etc) on those interfaces can then make further forwarding decisions with respect to the multicast packet. Note that the presence of multiple border routers associated with a MANET routing domain raises additional issues. This is further discussed in Section 9.4 but further work is expected to be needed here.

9.2. Multicast Group Scoping

Multicast scoping is used by network administrators to control the network routing domains reachable by multicast packets. This is usually done by configuring external interfaces of border routers in the border of a routing domain to not forward multicast packets which must be kept within the routing region. This is commonly done based on TTL of messages or the basis of group addresses. These schemes are known respectively as:

1. TTL scoping.
2. Administrative scoping.

For IPv4, network administrators can configure border routers with the appropriate TTL thresholds or administratively scoped multicast groups for the router interfaces as with any traditional multicast router. However, for the case of TTL scoping it SHOULD be taken into account that the packet could traverse multiple hops within the MANET SMF routing domain before reaching the border router. Thus, TTL thresholds SHOULD be selected carefully.

For IPv6, multicast address spaces include information about the scope of the group. Thus, border routers of an SMF routing domain know if they must forward a packet based on the IPv6 multicast group address. For the case of IPv6, it is RECOMMENDED that a MANET SMF routing domain be designated a site-scoped multicast domain. Thus, all IPv6 site-scoped multicast packets in the range FF05::/16 SHOULD be kept within the MANET SMF routing domain by border routers. IPv6 packets in any other wider range scopes (i.e. FF08::/16, FF0B::/16 and FF0E::16) MAY traverse border routers unless other restrictions different from the scope applies.

Given that scoping of multicast packets is performed at the border routers, and given that existing scoping mechanisms are not designed to work with mobile routers, it is assumed that non-border routers running SMF will not stop forwarding multicast data packets of an appropriate site scoping. That is, it is assumed that an SMF routing domain is a site-scoped multicast area.

9.3. Interface with Exterior Multicast Routing Protocols

The traditional operation of multicast routing protocols is tightly integrated with the group membership function. Leaf routers are configured to periodically gather group membership information, while intermediate routers conspire to create multicast trees connecting routers with directly-connected multicast sources and routers with active multicast receivers. In the concrete case of SMF, border routers can be considered leaf routers. Mechanisms for multicast sources and receivers to interoperate with border routers over the multihop MANET SMF routing domain as if they were directly connected to the router need to be defined. The following issues need to be addressed:

1. A mechanism by which border routers gather membership information
2. A mechanism by which multicast sources are known by the border router
3. A mechanism for exchange of exterior routing protocol messages across the SMF routing domain if the SMF routing domain is to provide transit connectivity for multicast traffic.

It is beyond the scope of this document to address implementation solutions to these issues. As described in Section 9.1, IGMP/MLD proxy mechanisms can be deployed to address some of these issues. Similarly, exterior routing protocol messages could be tunneled or conveyed across an SMF routing domain but doing this robustly in a distributed wireless environment likely requires additional considerations outside the scope of this document.

The need for the border router to receive traffic from recognized multicast sources within the SMF routing domain is important to potentially achieve interoperability with existing routing protocols. For instance, PIM-S requires routers with locally attached multicast sources to register them to the Rendezvous Point (RP) so that nodes can join the multicast tree. In addition, if those sources are not advertised to other autonomous systems (AS) using Multicast Source Discovery Protocol (MSDP), receivers in those external networks are not able to join the multicast tree for that source.

9.4. Multiple Border Routers

An SMF domain might be deployed with multiple participating nodes having connectivity to external, fixed-infrastructure networks. Allowing multiple nodes to forward multicast traffic to/from the SMF routing domain can be beneficial since it can increase reliability, and provide better service. For example, if the SMF routing domain were to fragment with different SMF nodes maintaining connectivity to different border routers, multicast service could still continue successfully. But, the case of multiple border routers connecting a SMF routing domain to external networks presents several challenges for SMF:

1. Handling duplicate unmarked IPv4 or IPv6 (without IPsec encapsulation or DPD option) packets possibly injected by multiple border routers.

2. Source-based relay algorithms handling of duplicate traffic injected by multiple border routers.
3. Determination of which border router(s) will forward outbound multicast traffic.
4. Additional challenges with interfaces to exterior multicast routing protocols.

When multiple border routers are present they may be alternatively (due to route changes) or simultaneously injecting common traffic into the MANET routing region that has not been previously marked for SMF-DPD. Different border routers would not be able to implicitly synchronize sequencing of injected traffic since they may not receive exactly the same messages due to packet losses. For IPv6 I-DPD operation, the optional "TaggerId" field described for the SMF-DPD header option can be used to mitigate this issue. When multiple border routers are injecting a flow into a MANET routing region, there are two forwarding policies that SMF nodes running I-DPD may implement:

1. Redundantly forward the multicast flows (identified by <srcAddr: dstAddr>) from each border router, performing DPD processing on a <taggerID:dstAddr> or <taggerID:srcAddr:dstAddr> basis, or
2. Use some basis to select the flow of one tagger (border router) over the others and forward packets for applicable flows (identified by <sourceAddress:dstAddr>) only for the selected "Tagger ID" until timeout or some other criteria to favor another tagger occurs.

It is RECOMMENDED that the first approach be used in the case of I-DPD operation. Additional specification may be required to describe an interoperable forwarding policy based on this second option. Note that the implementation of the second option requires that per-flow (i.e., <srcAddr::dstAddr>) state be maintained for the selected "Tagger ID".

The deployment of H-DPD operation may alleviate DPD resolution when ingressing traffic comes from multiple border routers. Non-colliding hash indexes (those not requiring the H-DPD options header in IPv6) should be resolved effectively.

10. Security Considerations

Gratuitous use of option headers can cause problems in routers. Other IP routers external to an SMF routing domains that might receive forwarded multicast should ignore SMF-specific header options when encountered. The header options types are encoded appropriately to allow for this behavior.

Here we briefly discuss several SMF denial-of-service (DoS) attack scenarios and we provide some initial recommended mitigation strategies.

A potential denial-of-service attack against SMF forwarding is possible when a malicious node has a form of wormhole access to multiple part of a network topology. In the wireless ad hoc case, a directional antenna is one way to provide such a wormhole physically. If such a node can preview forwarded packets in one part of the network and forward modified versions to another part of the network it can perform the following attack. The malicious node could reduce the TTL or Hop Limit of the packet and transmit it to the SMF node causing it to forward the packet with a limited TTL (or even drop it) and make a DPD entry that could block or limit the subsequent forwarding of later-arriving valid packets with correct TTL values. This would be a relatively low-cost, high-payoff attack that would be hard to detect and thus attractive to potential attackers. An approach of caching TTL information with DPD state and taking appropriate forwarding actions is identified in Section 5 to mitigate this form of attack.

Sequence-based packet identifiers are predictable and thus provide an opportunity for a DoS attack against forwarding. Forwarding protocols that use DPD techniques, such as SMF, may be vulnerable to DoS attacks based on spoofing packets with apparently valid packet identifier fields. In wireless environments, where SMF will most likely be used, the opportunity for such attacks may be more prevalent than in wired networks. In the case of IPv4 packets, fragmented IP packets or packets with IPsec headers applied, the DPD "identifier portions" of potential future packets that might be forwarded is highly predictable and easily subject to denial-of-service attacks against forwarding. A RECOMMENDED technique to counter this concern is for SMF implementations to generate an "internal" hash value that is concatenated with the explicit I-DPD packet identifier to form a unique identifier that is a function of the packet content as well as the visible identifier. SMF implementations could seed their hash generation with a random value to make it unlikely that an external observer could guess how to spoof packets used in a denial-of-service attack against forwarding. Since the hash computation and state is kept completely internal to SMF nodes, the cryptographic properties of this hashing would not need to be extensive and thus possibly of low complexity. Experimental implementations may determine that a lightweight hash of even only portions of packets may suffice to serve this purpose.

While H-DPD is not as readily susceptible to this form of DoS attack, it is possible that a sophisticated adversary could use side information to construct spoofing packets to mislead forwarders using a well-known hash algorithm. Thus, similarly, a separate "internal" hash value could be concatenated with the well-known hash value to alleviate this security concern.

The support of forwarding IPsec packets without further modification for both IPv4 and IPv6 is supported by this specification.

Authentication mechanisms to identify the source of IPv6 option headers should be considered to reduce vulnerability to a variety of attacks.

11. IANA Considerations

This document raises multiple IANA Considerations. These include the IPv6 SMF_DPD hop-by-hop Header Extension defined and multiple Type-Length-Value (TLV) constructs [RFC5444]) to be used with NHDP [RFC6130]operation as needed to support different forms of SMF operation. There is one message TLV type and one address TLV type needed to be assigned for SMF purposes as discussed in Section 8.1.

The value of the IPv6 SMF-DPD Hop-by-Hop Option Type is TBD (to be assigned).

The SL-MANET-ROUTERS multicast address will be registered for both IPv4 and IPv6 multicast address spaces.

11.1. IPv6 SMF-DPD Header Extension

This document requests IANA assignment of the "SMF_DPD" hop-by-hop option type from the IANA "IPv6 Hop-by-Hop Options Option Type" registry (see Section 5.5 of [RFC2780]).

The format of this new option type is described in Section 6.1.1. A portion of the option data content is the taggger identifier type "TidType" that provides a context for the "TaggerId" that is optionally included to identify the node that added the SMF_DPD option to the packet. This document defines a namespace for IPv6 SMF_DPD Tagger Identifier Type values:

ietf:manet:smf:taggerIdTypes

The values that can be assigned within the "ietf:manet:smf:taggerIdTypes" name-space are numeric indexes in the range [0, 7], boundaries included. All assignment requests are granted on an "IETF Consensus" basis as defined in [RFC5226].

This specification registers Tagger Identification Type values from Table 9 in the registry "ietf:manet:smf:taggerIdTypes":

Mnemonic	Value	Reference
NULL	0	This document
DEFAULT	1	This document
IPv4	2	This document
IPv6	3	This document
ExtId	7	This document

Table 9: TaggerId Types

11.2. SMF Type-Length-Value

This document requests IANA assignment of one message "SMF_TYPE" TLV type and one address block "SMF_NBR_TYPE" TLV type from the [RFC6130] specific registry space.

The common format of these new TLV types is described in Table 6 and Table 8. Furthermore this document defines a namespace for algorithm ID types using the extended type TLV value field defined by [RFC5444]. Both SMF_TYPE and SMF_NBR_TYPE TLVs use this namespace.

ietf:manet:packetbb:nhdp:smf:relayAlgorithmID

The values that can be assigned within the "ietf:manet:packetbb:nhdp: smf:relayAlgorithmID" name-space are numeric indexes in the range [0, 239], boundaries included. Assignment requests for the [0-127] are granted on an "IETF Consensus" basis as defined in [RFC5226]. Standards action is not required for assignment requests of the range [128-239]. Documents requesting relayAlgorithmId values SHOULD define value field uses contained by the SMF_TYPE:<relayAlgorithmId> and SMF_NBR_TYPE:<relayAlgorithmId> full type TLVs.

This specification registers the following Relay Algorithm ID Type values shown in Table 10 in the registry "ietf:manet:packetbb:nhdp: smf:relayAlgorithmID

Mnemonic	Value	Reference
CF	0	
S-MPR	1	Appendix B
E-CDS	2	Appendix A
MPR-CDS	3	Appendix C

Table 10: Relay Set Algorithm Type Values

12. Acknowledgments

Many of the concepts and mechanisms used and adopted by SMF resulted from many years of discussion and related work within the MANET working group since the late 1990s. There are obviously many contributors to past discussions and related draft documents within the working group that have influenced the development of SMF concepts that deserve acknowledgment. In particular, the document is largely a direct product of the earlier SMF design team within the IETF MANET working group and borrows text and implementation ideas from the related individuals and activities. Some of the direct contributors who have been involved in design, content editing, prototype implementation, major commenting, and core discussions are listed below in alphabetical order. We appreciate all the input and feedback from the many community members and early implementation users we have heard from that are not on this list as well.

Key contributors/authors in alphabetical order:
Brian Adamson
Teco Boot
Ian Chakeres
Thomas Clausen
Justin Dean
Brian Haberman
Ulrich Herberg
Charles Perkins
Pedro Ruiz
Fred Templin
Maoyu Wang

The RFC text was produced using Marshall Rose's xml2rfc tool and Bill Fenner's XMLmind add-ons.

13. References

13.1. Normative References

[E-CDS] Ogier, R., "MANET Extension of OSPF Using CDS Flooding", Proceedings of the 62nd IETF , March 2005.

[MPR-CDS] Adjih, C., Jacquet, P., and L. Viennot, "Computing Connected Dominating Sets with Multipoint Relays", Ad Hoc and Sensor Wireless Networks , January 2005.

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[RFC1321] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[RFC2644] Senie, D., "Changing the Default for Directed Broadcasts in Routers", BCP 34, RFC 2644, August 1999.

[RFC2780] Bradner, S., "IANA Allocation Guidelines For Values In the Internet Protocol and Related Headers", March 2000.

[RFC3626] Clausen, T. and P. Jacquet, "Optimized Link State Routing Protocol", 2003.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

[RFC4302] Kent, S., "IP Authentication Header", December 2005.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5444] Clausen, T. and et al, "Generalized MANET Packet/Message Format", RFC 5444, February 2009.

[RFC5771] Cotton, M., Vegoda, L., and D. Meyer, "IANA Guidelines for IPv4 Multicast Address Assignment", RFC 5771, March 2010.

[RFC6130] Clausen, T. and et al, "MANET Neighborhood Discovery Protocol", RFC 6130, March 2011.

13.2. Informative References

[CDHM07] Chakeres, I., Danilov, C., and T. Henderson, "Connecting MANET Multicast", IEEE MILCOM 2007 Proceedings , 2007.

[DHG09] Danilov, C., Henderson, T., and T. Goff, "Experiment and field demonstration of a 802.11-based ground-UAV mobile ad-hoc network", Proceedings of the 28th IEEE conference on Military Communications , 2009.

[DHS08] Danilov, C., Henderson, T., and T. Spagnolo, "MANET Multicast with Multiple Gateways", IEEE MILCOM 2008

Proceedings , 2008.

[GM99] Garcia-Luna-Aceves, JJ. and E. Madruga, "The core-assisted mesh protocol", Selected Areas in Communications, IEEE Journal on Volume 17, Issue 8, August 1999.

[JLMV02] Jacquet, P., Laouiti, V., Minet, P., and L. Viennot, "Performance of multipoint relaying in ad hoc mobile routing protocols", Networking , 2002.

[MDC04] Macker, J., Dean, J., and W. Chao, "Simplified Multicast Forwarding in Mobile Ad hoc Networks", IEEE MILCOM 2004 Proceedings , 2004.

[MDDA07] Macker, J., Downard, I., Dean, J., and R. Adamson, "Evaluation of distributed cover set algorithms in mobile ad hoc network for simplified multicast forwarding", ACM SIGMOBILE Mobile Computing and Communications Review Volume 11 , Issue 3, July 2007.

[MGL04] Mohapatra, P., Gui, C., and J. Li, "Group Communications in Mobile Ad hoc Networks", IEEE Computer Vol. 37, No. 2, February 2004.

[NTSC99] Ni, S., Tseng, Y., Chen, Y., and J. Sheu, "The Broadcast Storm Problem in Mobile Ad hoc Networks", Proceedings Of ACM Mobicom 99 , 1999.

[RFC2501] Macker, JP. and MS. Corson, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations", 1999.

[RFC3684] Ogier, R., Templin, F., and M. Lewis, "Topology Dissemination Based on Reverse-Path Forwarding", 2003.

[RFC3973] Adams, A., Nicholas, J., and W. Siadak, "Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised)", RFC 3973, January 2005.

[RFC4601] Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, August 2006.

Appendix A. Essential Connecting Dominating Set (E-CDS) Algorithm

The "Essential Connected Dominating Set" (E-CDS) algorithm [E-CDS] forms a single CDS mesh for the SMF operating region. It allows nodes to use 2-hop neighborhood topology information to dynamically perform relay self election to form a CDS. Its packet forwarding rules are not dependent upon previous hop knowledge. Additionally, E-CDS SMF forwarders can be easily mixed without problems with CF SMF forwarders, even those not participating in NHDP. Another benefit is that packets opportunistically received from non-symmetric neighbors may be forwarded without compromising flooding efficiency or correctness. Furthermore, multicast sources not participating in NHDP may freely inject their traffic and any neighboring E-CDS relays will properly forward the traffic. The E-CDS based relay set selection algorithm is based upon the summary within [E-CDS]. E-CDS was originally discussed in the context of forming partial adjacencies and efficient flooding for MANET OSPF extensions work and the core algorithm is applied here for SMF.

It is RECOMMENDED that the SMF_TYPE:E-CDS message TLV be included in NHDP_HELLO messages that are generated by nodes conducting E-CDS SMF operation. It is also RECOMMENDED that the SMF_NBR_TYPE:E-CDS address block TLV be used to advertise neighbor nodes that are also conducting E-CDS SMF operation.

A.1. E-CDS Relay Set Selection Overview

The E-CDS relay set selection requires 2-hop neighborhood information collected through NHDP or another process. Relay nodes, in E-CDS SMF selection, are "self-elected" using a router identifier (Router ID) and an optional nodal metric, referred to here as "Router Priority" for all 1-hop and 2-hop neighbors. To ensure proper relay set self-election, the Router ID and Router Priority MUST be consistent among participating nodes. It is RECOMMENDED that NHDP be used to share Router ID and Router Priority through the use of SMF_TYPE:E-CDS TLVs as described in this appendix.. The Router ID is a logical identification that MUST be consistent across interoperating SMF neighborhoods and it is RECOMMENDED to be chosen as the numerically largest address contained in a nodes "Neighbor Address List" as defined in NHDP. The E-CDS self-election process can be summarized as follows:

1. If an SMF node has a higher ordinal (Router Priority, Router ID) than all of its symmetric neighbors, it elects itself to act as a forwarder for all received multicast packets,
2. Else, if there does not exist a path from the neighbor with largest (Router Priority, Router ID) to any other neighbor, _via_ neighbors with larger values of (Router Priority, Router ID), then it elects itself to the relay set.

The basic form of E-CDS described and applied within this specification does not provide for redundant relay set election (e.g., bi-connected) but such capability is supported by the basic E-CDS design.

A.2. E-CDS Forwarding Rules

With E-CDS, any SMF node that has selected itself as a relay performs DPD and forwards all non-duplicative multicast traffic allowed by the present forwarding policy. Packet previous hop knowledge is not needed for forwarding decisions when using E-CDS.

1. Upon packet reception, DPD is performed. Note E-CDS requires a single duplicate table for the set of interfaces associated with the relay set selection.
2. If the packet is a duplicate, no further action is taken.
3. If the packet is non-duplicative:
   A. A DPD entry is made for the packet identifier
   B. The packet is forwarded out all interfaces associated with the relay set selection

As previously mentioned, even packets sourced (or relayed) by nodes not participating in NHDP and/or the E-CDS relay set selection may be forwarded by E-CDS forwarders without problem. A particular deployment MAY choose to not forward packets from previous hop nodes that have been not explicitly identified via NHDP or other means as operating as part of a different relay set algorithm (e.g. S-MPR) to allow coexistent deployments to operate correctly. Also, E-CDS relay set selection may be configured to be influenced by statically-configured CF relays that are identified via NHDP or other means.

A.3. E-CDS Neighborhood Discovery Requirements

It is possible to perform E-CDS relay set selection without modification of NHDP, basing the self-election process exclusively on the "Neighbor Address List" of participating SMF nodes. For example by setting the "Router Priority" to a default value and selecting the "Router ID" as the numerically largest address contained in the "Neighbor Address List". However steps MUST be taken to insure that all NHDP enabled nodes not using SMF_TYPE:E-CDS full type message TLVs are in fact running SMF E-CDS with the same methods for selecting "Router Priority" and "Router ID", otherwise incorrect forwarding may occur. Note that SMF nodes with higher "Router Priority" values will be favored as relays over nodes with lower "Router Priority". Thus, preferred relays MAY be administratively configured to be selected when possible. Additionally, other metrics (e.g. nodal degree, energy capacity, etc) may also be taken into account in constructing a "Router Priority" value. When using "Router Priority" with multiple interfaces all interfaces on a node MUST use and advertise a common "Router Priority" value. A nodes

"Router Priority" value may be administratively or algorithmically selected. The method of selection does not need to be the same among different nodes.

E-CDS relay set selection may be configured to be influenced by statically configured CF relays that are identified via NHDP or other means. Nodes advertising CF through NHDP may be considered E-CDS SMF nodes with maximal "Router Priority".

To share a node's "Router Priority" with its 1-hop neighbors the SMF_TYPE:E-CDS message TLV's <value> field is defined as shown in Table 11.

Length(bytes)	Value	Router Priority
0	N/A	64
1	<value>	0-127

Table 11: E-CDS Message TLV Values

Where <value> is a one octet long bit field which is defined as:

bit 0: the leftmost bit is reserved and SHOULD be set to 0.

bit 1-7: contain the unsigned "Router Priority" value, 0-127, which is associated with the "Neighbor Address List".

Combinations of value field lengths and values other than specified here are NOT permitted and SHOULD be ignored. Below is an example SMF_TYPE:E-CDS message TLV

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
               ...              |   SMF_TYPE    |1|0|0|1|0|0|   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     E-CDS     |0|0|0|0|0|0|0|1|R|  priority   |      ...      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: E-CDS Message TLV Example

To convey "Router Priority" values among 2-hop neighborhoods the SMF_NBR_TYPE:E-CDS address block TLV's <value> field is used. Multi-index and multi-value TLV layouts as defined in [RFC5444] are supported. SMF_NBR_TYPE:E-CDS value fields are defined thus:

Length(bytes)	# Addr	Value	Router Priority
0	Any	N/A	64
1	Any	<value>	<value> is for all addresses
N	N	<value>*	Each address gets its own <value>

Table 12: E-CDS Address Block TLV Values

Where <value> is a one byte bit field which is defined as:

bit 0: the leftmost bit is reserved and SHOULD be set to 0.

bit 1-7: contain the unsigned "Router Priority" value, 0-127, which is associated with the appropriate address(es).

Combinations of value field lengths and # of addresses other than specified here are NOT permitted and SHOULD be ignored. A default technique of using nodal degree (i.e. count of 1-hop neighbors) is RECOMMENDED for the value field of these TLV types. Below are two example SMF_NBR_TYPE:E-CDS address block TLVs.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |              ...              | SMF_NBR_TYPE  |1|0|0|1|0|0|   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     E-CDS     |0|0|0|0|0|0|0|1|R|  priority   |      ...      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: E-CDS Address Block TLV Example 1

The single value example TLV, depicted in Figure 6 , specifies that all address(es) contained in the address block are running SMF using the E-CDS algorithm and all address(es) share the value field and therefore the same "Router Priority".

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
               ...              | SMF_NBR_TYPE  |1|0|1|1|0|1|   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     E-CDS     |  index-start  |   index-end   |    length     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|R|  priority0  |R|  priority1  |      ...      |R|  priorityN  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: E-CDS Address Block TLV Example 2

The example multivalued TLV, depicted in Figure 7, specifies that address(es) contained in the address block from index-start to index-end inclusive are running SMF using the E-CDS algorithm. Each address is associated with its own value byte and therefore its own "Router Priority".

A.4. E-CDS Selection Algorithm

This section describes an algorithm for E-CDS relay selection (self-election). The algorithm described uses 2-hop information. Note it is possible to extend this algorithm to use k-hop information with added computational complexity and mechanisms for sharing k-hop topology information that are not described in this document or within the NHDP specification. It should also be noted that this algorithm does not impose the "hop limit" bound described in [E-CDS] when performing the path search that is used for relay selection. However, the algorithm below could be easily augmented to accommodate this additional criterion. It is not expected that the "hop limit" bound will provide significant benefit to the algorithm defined in this appendix.

The tuple of "Router Priority" and "Router ID" is used in E-CDS relay set selection. Precedence is given to the "Router Priority" portion and the "Router ID" value is used as a tie-breaker. The evaluation of this tuple is referred to as "RtrPri(n)" in the description below where "n" references a specific node. Note it is possible that the "Router Priority" portion may be optional and the evaluation of "RtrPri()" be solely based upon the unique "Router ID". Since there MUST NOT be any duplicate "Router ID" values among SMF nodes, a comparison of RtrPri(n) between any two nodes will always be an inequality. The use of nodal degree for calculating "Router Priority" is RECOMMENDED as default and the largest IP address in the "Neighbor Address List" as advertised by NHDP MUST be used as the "Router ID". NHDP provides all interface address throughout the 2-hop neighborhood through HELLO messages, so explicitly conveying a "Router ID" is not necessary. The following steps describe a basic algorithm for conducting E-CDS relay selection for a node "n0":

1. Initialize the set "N1" with tuples ("Router Priority", "Router ID", "Neighbor Address List" for each 1-hop neighbor of "n0".
2. If "N1" has less than 2 tuples, then "n0" does not elect itself as a relay and no further steps are taken.
3. Initialize the set "N2" with tuples ("Router Priority", "Router ID", "2-hop address") for each "2-hop address" of "n0", where "2-hop address" is defined in NHDP.
4. If "RtrPri(n0)" is greater than that of all tuples in the union of "N1" and "N2", then "n0" selects itself as a relay and no further steps are taken.
5. Initialize all tuples in the union of "N1" and "N2" as "unvisited".
6. Find the tuple "n1_Max" that has the largest "RtrPri()" of all tuples in "N1"
7. Initialize queue "Q" to contain "n1_Max", marking "n1_Max" as "visited"
8. While node queue "Q" is not empty, remove node "x" from the head of "Q", and for each 1-hop neighbor "n" of node "x" (excluding "n0") that is not marked "visited"
    A. Mark node "n" as "visited"
    B. If "RtrPri(n)" is greater than "RtrPri(n0), append "n" to "Q"
9. If any tuple in "N1" remains "unvisited", then "n0" selects itself as a relay. Otherwise "n0" does not act as a relay.

Note these steps are re-evaluated upon neighborhood status changes. Steps 5 through 8 of this procedure describe an approach to a path search. The purpose of this path search is to determine if paths exist from the 1-hop neighbor with maximum "RtrPri()" to all other 1-hop neighbors without traversing an intermediate node with a "RtrPri()" value less than "RtrPri(n0)". These steps comprise a breadth-first traversal that evaluates only paths that meet that criteria. If all 1-hop neighbors of "n0" are "visited" during this traversal, then the path search has succeeded and node "n0" does not need to provide relay. It can be assumed that other nodes will provide relay operation to ensure SMF connectivity.

It is possible to extend this algorithm to consider neighboring SMF nodes that are known to be statically configured for CF (always relaying). The modification to the above algorithm is to process such nodes as having a maximum possible "Router Priority" value. It is expected that nodes configured for CF and participating in NHDP would indicate this with use of the SMF_TYPE:CF and SMF_NBR_TYPE:CF TLV types in their NHDP_HELLO message and address blocks, respectively.

Appendix B. Source-based Multipoint Relay (S-MPR)

The source-based multipoint relay (S-MPR) set selection algorithm enables individual nodes, using two-hop topology information, to select relays from their set of neighboring nodes. Relays are selected so that forwarding to the node's complete two-hop neighbor set is covered. This distributed relay set selection technique has been shown to approximate a minimal connected dominating set (MCDS) in [JLMV02]. Individual nodes must collect two-hop neighborhood information from neighbors, determine an appropriate current relay set, and inform selected neighbors of their relay status. Note that since each node picks its neighboring relays independently, S-MPR forwarders depend upon previous hop information (e.g, source MAC address) to operate correctly. The Optimized Link State Routing (OLSR) protocol has used this algorithm and protocol for relay of link state updates and other control information [RFC3626] and it has been demonstrated operationally in dynamic network environments.

It is RECOMMENDED that the SMF_TYPE:S-MPR message TLV be included in NHDP_HELLO messages that are generated by nodes conducting S-MPR SMF operation. It is also RECOMMENDED that the SMF_NBR_TYPE:S-MPR address block TLV be used to specify which neighbor nodes are conducting S-MPR SMF operation.

B.1. S-MPR Relay Set Selection Overview

The S-MPR algorithm uses bi-directional 1-hop and 2-hop neighborhood information collected via NHDP to select, from a node's 1-hop neighbors, a set of relays that will cover the node's entire 2-hop neighbor set upon forwarding. The algorithm described uses a "greedy" heuristic of first picking the 1-hop neighbor who will cover the most 2-hop neighbors. Then, excluding those 2-hop neighbors that have been covered, additional relays from its 1-hop neighbor set are iteratively selected until the entire 2-hop neighborhood is covered. Note that 1-hop neighbors also identified as 2-hop neighbors are considered as 1-hop neighbors only.

NHDP HELLO messages supporting S-MPR forwarding operation SHOULD use the TLVs defined in Section 8.1 using the S-MPR extended type. The value field of an address block TLV which has a full type value of SMF_NBR_TYPE:S-MPR is defined in Table 14 such that signaling of MPR selections to 1-hop neighbors is possible. The value field of a message block TLV which has a full type value of SMF_TYPE:S-MPR is defined in Table 13 such that signaling of "Router Priority" (described as "WILLINGNESS" in [RFC3626]) to 1-hop neighbors is possible. It is important to note that S-MPR forwarding is dependent upon the previous hop of an incoming packet. An S-MPR node MUST forward packets only for neighbors which have explicitly selected it as a multi-point relay (i.e., its "selectors"). There are also some additional requirements for duplicate packet detection to support S-MPR SMF operation that are described below.

For multiple interface operation, MPR selection SHOULD be conducted on a per-interface basis. However, it is possible to economize MPR selection among multiple interfaces by selecting common MPRs to the extent possible.

B.2. S-MPR Forwarding Rules

An S-MPR SMF node MUST only forward packets for neighbors that have explicitly selected it as an MPR. The source-based forwarding technique also stipulates some additional duplicate packet detection operations. For multiple network interfaces, independent DPD state MUST be maintained for each separate interface. The following table provides the procedure for S-MPR packet forwarding given the arrival of a packet on a given interface, denoted <srcIface>. There are three possible actions, depending upon the previous-hop transmitter:

1. If the previous-hop transmitter has selected the current node as an MPR,
   A. The packet identifier is checked against the DPD state for each possible outbound interface, including the <srcIface>.
   B. If the packet is not a duplicate for an outbound interface, the packet is forwarded on that interface and a DPD entry is made for the given packet identifier for the interface.
   C. If the packet is a duplicate, no action is taken for that interface.
2. Else, if the previous-hop transmitter is a 1-hop symmetric neighbor,
   A. A DPD entry is added for that packet for the <srcIface>, but the packet is not forwarded.
3. Otherwise, no action is taken.

Case number two in the above table is non-intuitive, but important to ensure correctness of S-MPR SMF operation. The selection of source-based relays does not result in a common set among neighboring nodes, so relays MUST mark in their DPD state, packets received from non-selector, symmetric, one-hop neighbors (for a given interface) and not forward subsequent duplicates of that packet if received on that interface. Deviation here can result in unnecessary, repeated packet forwarding throughout the network, or incomplete flooding.

Nodes not participating in neighborhood discovery and relay set selection will not be able to source multicast packets into the area and have SMF forward them, unlike E-CDS or MPR-CDS where forwarding may occur dependent on topology. Correct S-MPR relay behavior will occur with the introduction of repeaters (non-NHDP/SMF participants that relay multicast packets using duplicate detection and CF) but the repeaters will not efficiently contribute to S-MPR forwarding as these nodes will not be identified as neighbors (symmetric or otherwise) in the S-MPR forwarding process. NHDP/SMF participants MUST NOT provide extra forwarding, forwarding packets which are not selected by the algorithm, as this can disrupt network-wide S-MPR flooding, resulting in incomplete or inefficient flooding. The result is that non S-MPR SMF nodes will be unable to source multicast packets and have them forwarded by other S-MPR SMF nodes.

B.3. S-MPR Neighborhood Discovery Requirements

Nodes may optionally signal a "Router Priority" value to their one hop neighbors by using the SMF_TYPE:S-MPR message block TLV value field. If the value field is omitted, a default "Router Priority" value of 64 is to be assumed. This is summarized here:

Length(bytes)	Value	Router Priority
0	N/A	64
1	<value>	0-127

Table 13: S-MPR Message TLV Values

Where <value> is a one octet long bit field defined as:

bit 0: the leftmost bit is reserved and SHOULD be set to 0.

bit 1-7: contain the "Router Priority" value, 0-127, which is associated with the "Neighbor Address List".

"Router Priority" values for S-MPR are interpreted in the same fashion as "WILLINGNESS" ([RFC3626])with value 0 indicating a node will NEVER forward and value 127 indicating a node will ALWAYS forward. Values 1-126 indicate how likely a S-MPR SMF router will be selected as an MPR by a neighboring SMF node, with higher values increasing the likelihood. Combinations of value field lengths and values other than specified here are NOT permitted and SHOULD be ignored. Below is an example SMF_TYPE:S-MPR message TLV.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
               ...               |   SMF_TYPE    |1|0|0|1|0|0|   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     S-MPR     |0|0|0|0|0|0|0|1|R|  priority     |     ...       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8: S-MPR Message TLV Example

S-MPR election operation requires 2-hop neighbor knowledge as provided by the NHDP protocol [RFC6130] or from external sources. MPRs are dynamically selected by each node and selections MUST be advertised and dynamically updated within NHDP or an equivalent protocol or mechanism. For NHDP use, the SMF_NBR_TYPE:S-MPR address block TLV value field is defined as such:

Length(bytes)	# Addr	Value	Meaning
0	Any	N/A	NOT MPRs
1	Any	<value>	<value> is for all addresses
N	N	<value>*	Each address gets its own <value>

Table 14: S-MPR Address Block TLV Values

Where <value>, if present, is a one octet bit field defined as:

bit 0: The leftmost bit is the M bit. When set indicates MPR selection of the relevant interface, represented by the associated address(es), by the originator node of the NHDP HELLO message. When unset, indicates the originator node of the NHDP HELLO message has not selected the relevant interfaces, represented by the associated address(es), as its MPR.

bit 1-7: are reserved and SHOULD be set to 0.

Combinations of value field lengths and number of addresses other than specified here are NOT permitted and SHOULD be ignored. All bits, excepting the leftmost bit, are RESERVED and SHOULD be set to 0.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              ...              | SMF_NBR_TYPE  |1|1|0|1|0|0|   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     S-MPR     |  start-index  |0|0|0|0|0|0|0|1|M|  reserved   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9: S-MPR Address Block TLV Example

The single index TLV example, depicted in Figure 9, indicates that the address specified by the <start-index> field is running SMF using S-MPR and has been selected by the originator of the NHDP HELLO message as an MPR forwarder if the M bit is set. Multivalued TLVs may also be used to specify MPR selection status of multiple addresses using only one TLV. See Figure 7 for a similar example on how this may be done.

B.4. S-MPR Selection Algorithm

This section describes a basic algorithm for the S-MPR selection process. Note that the selection is with respect to a specific interface of the node performing selection and other node interfaces referenced are reachable from this reference node interface. This is consistent with the S-MPR forwarding rules described above. When multiple interfaces per node are used, it is possible to enhance the overall selection process across multiple interfaces such that common nodes are selected as MPRs for each interface to avoid unnecessary inefficiencies in flooding. The following steps describe a basic algorithm for conducting S-MPR selection for a node interface "n0":

1. Initialize the set "MPR" to empty.
2. Initialize the set "N1" to include all 1-hop neighbors of "n0".
3. Initialize the set "N2" to include all 2-hop neighbors, excluding "n0" and any nodes in "N1". Nodes which are only reachable via "N1" nodes with router priority values of NEVER are also excluded.
4. For each interface "y" in "N1", initialize a set "N2(y)" to include any interfaces in "N2" that are 1-hop neighbors of "y".
5. For each interface "x" in "N1" with a router priority value of "ALWAYS" (or using CF relay algorithm), select "x" as a MPR:
   A. Add "x" to the set "MPR" and remove "x" from "N1".
   B. For each interface "z" in "N2(x)", remove "z" from "N2"
   C. For each interface "y" in "N1", remove any interfaces in "N2(x)" from "N2(y)"
6. For each interface "z" in "N2", initialize the set "N1(z)" to include any interfaces in "N1" that are 1-hop neighbors of "z".

7. For each interface "x" in "N2" where "N1(x)" has only one member, select "x" as a MPR:
   A. Add "x" to the set "MPR" and remove "x" from "N1".
   B. For each interface "z" in "N2(x)", remove "z" from "N2" and delete "N1(z)"
   C. For each interface "y" in "N1", remove any interfaces in "N2(x)" from "N2(y)"
8. While "N2" is not empty, select the interface "x" in "N1" with the largest router priority which has the number of members in "N_2(x)" as a MPR:
   A. Add "x" to the set "MPR" and remove "x" from "N1".
   B. For each interface "z" in "N2(x)", remove "z" from "N2"
   C. For each interface "y" in "N1", remove any interfaces in "N2(x)" from "N2(y)"

After the set of nodes "MPR" is selected, node "n_0" must signal its selections to its neighbors. With NHDP, this is done by using the MPR address block TLV to mark selected neighbor addresses in NHDP_HELLO messages. Neighbors MUST record their MPR selection status and the previous hop address (e.g., link or MAC layer) of the selector. Note these steps are re-evaluated upon neighborhood status changes.

Appendix C. Multipoint Relay Connected Dominating Set (MPR-CDS) Algorithm

The MPR-CDS algorithm is an extension to the basic S-MPR election algorithm that results in a shared (non source-specific) SMF CDS. Thus its forwarding rules are not dependent upon previous hop information similar to E-CDS. An overview of the MPR-CDS selection algorithm is provided in [MPR-CDS].

It is RECOMMENDED that the SMF_TYPE Message TLV be included in NHDP_HELLO messages that are generated by nodes conducting MPR-CDS SMF operation.

C.1. MPR-CDS Relay Set Selection Overview

The MPR-CDS relay set selection process is based upon the MPR selection process of the S-MPR algorithm with the added refinement of a distributed technique for subsequently down-selecting to a common reduced, shared relay set. A node ordering (or "prioritization") metric is used as part of this down-selection process like the E-CDS algorithm, this metric can be based upon node address(es) or some other unique router identifier (e.g. "Router ID" based on largest address contained within the "Neighbor Address List") as well as an additional "Router Priority" measure, if desired. The process for

MPR-CDS relay selection is as follows:

1. First, MPR selection per the S-MPR algorithm is conducted, with selectors informing their MPRs (via NHDP) of their selection.
2. Then, the following rules are used on a distributed basis by selected nodes to possibly deselect themselves and thus jointly establish a common set of shared SMF relays:
   A. If a selected node has a larger "RtrPri()" than all of its 1-hop symmetric neighbors, then it acts as a relay for all multicast traffic, regardless of the previous hop
   B. Else, if the 1-hop symmetric neighbor with the largest "RtrPri()" value has selected the node, then it also acts as a relay for all multicast traffic, regardless of the previous hop.
   C. Otherwise, it deselects itself as a relay and does not forward any traffic unless changes occur that require re-evaluation of the above steps.

This technique shares many of the desirable properties of the E-CDS technique with regards to compatibility with multicast sources not participating in NHDP and the opportunity for statically-configure CF nodes to be present, regardless of their participation in NHDP.

C.2. MPR-CDS Forwarding Rules

The forwarding rules for MPR-CDS are common with those of E-CDS. Any SMF node that has selected itself as a relay performs DPD and forwards all non-duplicative multicast traffic allowed by the present forwarding policy. Packet previous hop knowledge is not needed for forwarding decisions when using MPR-CDS.

1. Upon packet reception, DPD is performed. Note MPR-CDS require one duplicate table for the set of interfaces associated with the relay set selection.
2. If the packet is a duplicate, no further action is taken.
3. If the packet is non-duplicative:
   A. A DPD entry is added for the packet identifier
   B. The packet is forwarded out all interfaces associated with the relay set selection

As previously mentioned, even packets sourced (or relayed) by nodes not participating in NHDP and/or the MPR-CDS relay set selection may be forwarded by MPR-CDS forwarders without problem. A particular deployment MAY choose to not forward packets from sources or relays that have been explicitly identified via NHDP or other means as operating as part of a different relay set algorithm (e.g. S-MPR) to allow coexistent deployments to operate correctly.

C.3. MPR-CDS Neighborhood Discovery Requirements

The neighborhood discovery requirements for MPR-CDS have commonality with both the S-MPR and E-CDS algorithms. MPR-CDS selection operation requires 2-hop neighbor knowledge as provided by the NHDP protocol [RFC6130] or from external sources. Unlike S-MPR operation, there is no need for associating link-layer address information with 1-hop neighbors since MPR-CDS forwarding is independent of the previous hop similar to E-CDS forwarding.

To advertise an optional "Router Priority" value or "WILLINGNESS" an originating node may use the message TLV of type SMF_TYPE:MPR-CDS which shares a common <value> format with both SMF_TYPE:E-CDS Table 11 and SMF_TYPE:S-MPR Table 13.

MPR-CDS only requires 1-hop knowledge of "Router Priority" for correct operation. In the S-MPR phase of MPR-CDS selection, MPRs are dynamically determined by each node and selections MUST be advertised and dynamically updated using NHDP or an equivalent protocol or mechanism. Therefore the <value> field of the SMF_NBR_TYPE:MPR-CDS type TLV shares a common format with SMF_NBR_TYPE:S-MPR Table 14 to convey MPR selection.

C.4. MPR-CDS Selection Algorithm

This section describes an algorithm for the MPR-CDS selection process. Note that the selection described is with respect to a specific interface of the node performing selection and other node interfaces referenced are reachable from this reference node interface. An ordered tuple of "Router Priority" and "Router ID" is used in MPR-CDS relay set selection. The "Router ID" value should be set to the largest advertised address of a given node, this information is provided to one hop neighbors via NHDP by default. Precedence is given to the "Router Priority" portion and the "Router ID" value is used as a tie-breaker. The evaluation of this tuple is referred to as "RtrPri(n)" in the description below where "n" references a specific node. Note it is possible that the "Router Priority" portion may be optional and the evaluation of "RtrPri()" be solely based upon the unique "Router ID". Since there MUST NOT be any duplicate address values among SMF nodes, a comparison of RtrPri(n) between any two nodes will always be an inequality. The following steps, repeated upon any changes detected within the 1-hop and 2-hop neighborhood, describe a basic algorithm for conducting MPR-CDS selection for a node interface "n0":

1. Perform steps 1-8 of Appendix B.4 to select MPRs from the set of 1-hop neighbors of "n0" and notify/update neighbors of selections.

2. Upon being selected as an MPR (or any change in the set of nodes selecting "n0" as an MPR):
   A. If no neighbors have selected "n0" as an MPR, "n0" does not act as a relay and no further steps are taken until a change in neighborhood topology or selection status occurs.
   B. Determine the node "n1_max" that has the maximum "RtrPri()" of all 1-hop neighbors.
   C. If "RtrPri(n0)" is greater than "RtrPri(n1_max)", then "n0" selects itself as a relay for all multicast packets,
   D. Else, if "n1_max" has selected "n0" as an MPR, then "0" selects itself as a relay for all multicast packets.
   E. Otherwise, "n0" does not act as a relay.

It is possible to extend this algorithm to consider neighboring SMF nodes that are known to be statically configured for CF (always relaying). The modification to the above algorithm is to process such nodes as having a maximum possible "Router Priority" value. This is the same as the case for participating nodes that have been configured with a S-MPR "WILLINGNESS" value of "WILL_ALWAYS". It is expected that nodes configured for CF and participating in NHDP would indicate their status with use of the SMF_TYPE TLV type in their NHDP_HELLO message TLV block. It is important to note however that CF nodes will not select MPR nodes and therefore cannot guarantee connectedness.

Authors' Addresses

Joseph Macker
NRL
Washington, DC 20375
USA

Email: macker@itd.nrl.navy.mil

SMF Design Team
IETF MANET WG

Email: manet@ietf.org

Internet Engineering Task Force R. Cole
Internet-Draft US Army CERDEC
Intended status: Standards Track J. Macker
Expires: July 20, 2011 B. Adamson
Naval Research Laboratory
S. Harnedy
Booz Allen Hamilton
January 16, 2011

Definition of Managed Objects for the Manet Simplified Multicast Framework Relay Set Process
draft-ietf-manet-smf-mib-02

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 objects for configuring aspects of the Simplified Multicast Forwarding (SMF) process for Mobile Ad-Hoc Networks (MANETs). The SMF-MIB also reports state information, performance metrics, and notifications. In addition to configuration, the additional state and performance information is useful to operators troubleshooting multicast forwarding problems.

Status of This Memo

Copyright Notice

Table of Contents

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 configuring aspects of a process implementing Simplified Multicast Forwarding (SMF) [I-D.ietf-manet-smf] for Mobile Ad-Hoc Networks (MANETs). SMF provides multicast Duplicate Packet Detection (DPD) and supports algorithms for constructing an estimate of a MANET Minimum Connected Dominating Set (MCDS) for efficient multicast forwarding. The SMF-MIB also reports state information, performance metrics, and notifications. In addition to configuration, this additional state and performance information is useful to operators troubleshooting multicast forwarding problems.

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. Conventions

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 [RFC2119].

4. Overview

SMF provides methods for implementing DPD-based multicast forwarding with the optional use of Connected Dominating Set (CDS)-based relay sets. The CDS provides a complete connected coverage of the nodes comprising the MANET. The MCDS is the smallest set of MANET nodes (comprising a connected cluster) which cover all the nodes in the cluster with their transmissions. As the density of the MANET nodes increase, the fraction of nodes required in an MCDS decreases. Using the MCDS as a multicast forwarding set then becomes an efficient multicast mechanism for MANETs.

Various algorithms for the construction of estimates of the MCDS exist. The Simplified Multicast Framework [I-D.ietf-manet-smf] describes some of these. It further defines various operational modes for a node which is participating in the collective creation of the MCDS estimates. These modes depend upon the set of related MANET routing and discovery protocols and mechanisms in operation in the specific MANET node.

A SMF router's MIB contains SMF process configuration parameters (e.g. specific CDS algorithm), state information (e.g., current membership in the CDS), performance counters (e.g., packet counters), and notifications.

4.1. SMF Management Model

This section describes the management model for the SMF node process.

Figure 1 (reproduced from Figure 4 of [I-D.ietf-manet-smf]) shows the relationship between the SMF Relay Set selection algorithm and the related algorithms, processes and protocols running in the MANET nodes. The Relay Set Selection Algorithm (RSSA) can rely upon topology information gotten from the MANET Neighborhood Discovery Protocol (NHDP), from the specific MANET routing protocol running on the node, or from Layer 2 information passed up to the higher layer protocol processes.

RGC Note: update this figure from the latest SMF draft.

```
       ______________                ____________
      |              |              |            |
      | Neighborhood |              | Relay Set  |
      |  Discovery   |------------->| Selection  |
      |   Protocol   |   neighbor   | Algorithm  |
      |______________|     info     |____________|
             \                               /
              \                             /
       neighbor\                           / forwarding
         info*  \      _____________      /    status
                 \    |             |    /
                  `-->| Forwarding  |<--'
                      |   Process   |
    ----------------->|_____________|----------------->
     incoming packet,                   forwarded packets
     interface id*, and
     previous hop*
```

Figure 1: SMF Node Architecture

4.2. Terms

The following definitions apply throughout this document:

- o Configuration Objects - switches, tables, objects which are initialized to default settings or set through the management interface defined by this MIB.
- o Tunable Configuration Objects - objects whose values affect timing or attempt bounds on the SMF RS process.
- o State Objects - automatically generated values which define the current operating state of the SMF RS process in the router.
- o Performance Objects - automatically generated values which help an administrator or automated tool to assess the performance of the CDS multicast process on the router and the overall multicasting performance within the MANET routing domain.

5. Structure of the MIB Module

This section presents the structure of the SMF-MIB module. The objects are arranged into the following groups:

- o smfMIBNotifications - defines the notifications associated with the SMF-MIB.

- o smfMIBObjects - defines the objects forming the basis for the SMF-MIB. These objects are divided up by function into the following groups:

- o

  * Capabilities Group - This group contains the SMF objects that the device uses to advertise its local capabilities with respect to, e.g., the supported RSSAs.

  * Configuration Group - This group contains the SMF objects that configure specific options that determine the overall operation of the SMF RSSA and the resulting multicast performance.

  * State Group - Contains information describing the current state of the SMF RSSA process such as the Neighbor Table.

  * Performance Group - Contains objects which help to characterize the performance of the SMF RSSA process, typically statistics counters.

- o smfMIBConformance - defines minimal and full conformance of implementations to this SMF-MIB.

5.1. Textual Conventions

The textual conventions defined within the SMF-MIB are as follows:

- o The SmfStatus is defined within the SMF-MIB. This contains the current operational status of the SMF process on an interface.

- o The SmfOpModeID represents an index that identifies a specific SMF operational mode.

- o The SmfRssaID represents an index that identifies, through reference, a specific RSSA available for operation on the device.

5.2. The Capabilities Group

The SMF device supports a set of capabilities. The list of capabilities which the device can advertise are:

- o Operational Mode - topology information from NHDP, CDS-aware unicast routing or Cross-layer from Layer 2.

- o SMF RSSA - the specific RSSA operational on the device. Note that configuration, state and performance objects related to a specific RSSA must be defined within another separate MIB.

5.3. The Configuration Group

The SMF device is configured with a set of controls. Some of the prominent configuration controls for the SMF device follow:

- o Operational Mode - topology information from NHDP, CDS-aware unicast routing or Cross-layer from Layer 2.

- o SMF RSSA - the specific RSSA operational on the device.

- o Duplicate Packet detection for IPv4 - Identification-based or Hash-based DPD.

- o Duplicate Packet detection for IPv6 - Identification-based or Hash-based DPD.

- o SMF Type Message TLV - if NHDP mode is selected, then is the SMF Type Message TLV may be included in the NHDP exchanges.

- o SMF Address Block TLV - if NHDP mode is selected, then is the SMF Address Block TLV included in the NHDP exchanges. (Note: is this correct?)

5.4. The State Group

The State Subtree reports current state information, e.g.,

- o Node RSS State - is the node currently in or out of the Relay Set.

- o Neighbors Table - a table containing current neighbors and their operational RSSA.

5.5. The Performance Group

The Performance subtree reports primarily counters that relate to SMF RSSA performance. The SMF performance counters consists of per node and per interface objects:

- o Total multicast packets received.

- o Total multicast packets forwarded.

- o Total duplicate multicast packets detected.

- o Per interface statistics table with the following entries:

- o

* Multicast packets received.

* Multicast packets forwarded.

* Duplicate multicast packets detected.

5.6. The Notifications Group

The Notifications Subtree contains the list of notifications supported within the SMF-MIB and their intended purpose or utility.

6. Relationship to Other MIB Modules

[TODO]: The text of this section specifies the relationship of the MIB modules contained in this document to other standards, particularly to standards containing other MIB modules. Definitions imported from other MIB modules and other MIB modules that SHOULD be implemented in conjunction with the MIB module contained within this document are identified in this section.

6.1. Relationship to the SNMPv2-MIB

The 'system' group in the SNMPv2-MIB [RFC3418] is defined as being mandatory for all systems, and the objects apply to the entity as a whole. The 'system' group provides identification of the management entity and certain other system-wide data. The SMF-MIB does not duplicate those objects.

6.2. MIB modules required for IMPORTS

The textual conventions imported for use in the SMF-MIB are as follows. The MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE, Counter32, Unsigned32, Integer32 and mib-2 textual conventions are imported from RFC 2578 [RFC2578]. The TEXTUAL-CONVENTION, RowStatus and TruthValue textual conventions are imported from RFC 2579 [RFC2579]. The MODULE-COMPLIANCE, OBJECT-GROUP and NOTIFICATION-GROUP textual conventions are imported from RFC 2580 [RFC2580]. The InterfaceIndexOrZero textual convention is imported from RFC 2863 [RFC2863]. The SnmpAdminString textual convention is imported from RFC 3411 [RFC3411]. The InetAddress, InetAddressType and InetAddressPrefixLength textual conventions are imported from RFC 4001 [RFC4001].

6.3. Relationship to the Future RSSA-MIBs

In a sense, the SMF-MIB is a general front-end to a set of, yet to be developed, RSSA-specific MIBs. These RSSA-specific MIBs will define the objects for the configuration, state, performance and notification objects required for the operation of these specific RSSAs. The SMF-MIB Capabilities Group allows the remote management station the ability to query the router to discover the set of supported RSSAs.

7. Definitions

```
MANET-SMF-MIB DEFINITIONS ::= BEGIN

IMPORTS

   MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
   Counter32, Unsigned32, Integer32, TimeTicks, mib-2
      FROM SNMPv2-SMI                          -- [RFC2578]

   TEXTUAL-CONVENTION, RowStatus, TruthValue
      FROM SNMPv2-TC                           -- [RFC2579]

   MODULE-COMPLIANCE, OBJECT-GROUP,
   NOTIFICATION-GROUP
      FROM SNMPv2-CONF                         -- [RFC2580]

   InterfaceIndexOrZero
      FROM IF-MIB                              -- [RFC2863]

   SnmpAdminString
      FROM SNMP-FRAMEWORK-MIB                  -- [RFC3411]

   InetAddress, InetAddressType,
   InetAddressPrefixLength
      FROM INET-ADDRESS-MIB                    -- [RFC4001]
   ;

manetSmfMIB MODULE-IDENTITY
   LAST-UPDATED "201101161300Z"  -- January 16, 2011
   ORGANIZATION "IETF MANET Working Group"
   CONTACT-INFO
      "WG E-Mail: manet@ietf.org

       WG Chairs: ian.chakeres@gmail.com
                  jmacker@nrl.navy.mil


       Editors:   Robert G. Cole
                  US Army CERDEC
                  Space and Terrestrial Communications
```

```
                     328 Hopkins Road
                     Bldg 245, Room 16
                     Aberdeen Proving Ground, MD 21005
                     USA
                     +1 410 278-6779
                     robert.g.cole@us.army.mil
                     http://www.cs.jhu.edu/~rgcole/

                     Joseph Macker
                     Naval Research Laboratory
                     Washington, D.C. 20375
                     USA
                     macker@itd.nrl.navy.mil

                     Brian Adamson
                     Naval Research Laboratory
                     Washington, D.C. 20375
                     USA
                     adamson@itd.nrl.navy.mil

                     Sean Harnedy
                     Booz Allen Hamilton
                     333 City Boulevard West
                     Orange, CA 92868
                     USA
                     +1 714 938-3898
                     harnedy_sean@bah.com"

      DESCRIPTION
         "This MIB module contains managed object definitions for
          the Manet SMF RSSA process defined in:

          [SMF] Macker, J.(ed.),
          Simplified Multicast Forwarding draft-ietf-manet-smf-10,
          March 06, 2010.

          Copyright (C) The IETF Trust (2008). This version
          of this MIB module is part of RFC xxxx; see the RFC
          itself for full legal notices."

        -- Revision History
        REVISION    "201101161300Z"   -- January 16, 2011
        DESCRIPTION
           "Updated 5th revision of the
            draft of this MIB module published as
            draft-ietf-manet-smf-mib-02.txt. The changes
            made in this revision include:
              - Added the Notification Group and cleaned
```

```
                up the Conformance section
              - Completed the TEXTUAL CONVENTION for the
                smfOpMode.
              - Completed the Description clauses of
                several objects within the MIB.
              - Removed the routerPriority object.
              - Added the definition of a smfRouterID
                object and associated smfRouterIDAddrType
                object.
            "
        REVISION    "200910261300Z"   -- October 26, 2009
        DESCRIPTION
           "Updated draft of this MIB module published as
            draft-ietf-manet-smf-mib-01.txt. A few changes
            were made in the development of this draft.
            Specifically, the following changes were made:
               - Updated the textual material, included
                 section on IMPORTS, relationship to other
                 MIBs, etc.
           "
        REVISION    "200904211300Z"   -- April 21, 2009
        DESCRIPTION
           "Updated draft of this MIB module published as
            draft-ietf-manet-smf-mib-00.txt. A few changes
            were made in the development of this draft.
            Specifically, the following changes were made:
               - Removed the smfGatewayFilterTable from this
                 draft.  It is a useful construct, e.g.,
                 an IPTABLES-MIB, but might best be handled
                 as a seperate MIB and worked within a
                 security focused working group.
               - Removed the smfReportsGroup. This capability
                 is being replaced with a new and more general
                 method for offline reporting.  This is being
                 worked as a new MIB module refered to as the
                 REPORT-MIB.
               - Rev'd as a new MANET WG document.
           "
        REVISION    "200902271300Z"   -- February 27, 2009
        DESCRIPTION
           "Updated draft of this MIB module published as
            draft-cole-manet-smf-mib-02.txt. Fairly extensive
            revisions and additions to this MIB were made
            in this version. Specifically, the following
            changes were made in development of this version:
               - added a Capabilities Group within the Objects
                 Group to allow the device to report supported
                 capabilities, e.g., RSSAs supported.
```

```
               - added administrative status objects for device
                 and interfaces
               - added multicast address forwarding tables, both
                 for configured (within Configuration Group) and
                 discovered (within the State Group).
               - added additional Performance counters related
                 to DPD functions.
               - Split up the performance counters into IPv4
                 and IPv6, for both global and per interface
                 statistics.
               - Split out the reports capability into a seperate
                 Reports Group under the Objects Group.
           "
        REVISION    "200811031300Z"   -- November 03, 2008
        DESCRIPTION
           "Updated draft of this MIB module published as
            draft-cole-manet-smf-mib-01.txt. Added gateway filter
            table and reports capabilities following rmon."
        REVISION    "200807071200Z"   -- July 07, 2008
        DESCRIPTION
           "Initial draft of this MIB module published as
            draft-cole-manet-smf-mib-00.txt."
        -- RFC-Editor assigns XXXX
        ::= { mib-2 998 }   -- to be assigned by IANA



   --
   -- TEXTUAL CONVENTIONs
   --

   SmfStatus ::= TEXTUAL-CONVENTION
       STATUS       current
       DESCRIPTION
          "An indication of the operability of a SMF
          function or feature.  For example, the status
          of an interface: 'enabled' indicates that
          it is performing SMF functions,
          and 'disabled' indicates that it is not."
       SYNTAX  INTEGER {
                        enabled (1),
                        disabled (2)
               }

   SmfOpModeID ::= TEXTUAL-CONVENTION
       STATUS       current
       DESCRIPTION
           "An index that identifies through reference to a specific
```

```
            SMF operations mode.  There are basically three styles
            of SMF operation with reduced relay sets:

              Independent operation - SMF performs its own relay
                  set selection using information from an associated
                  MANET NHDP process.

              CDS-aware unicast routing operation - a coexistent
                  unicast routing protocol provides dynamic relay
                  set state based upon its own control plane
                  CDS or neighborhood discovery information.

              Cross-layer operation -  SMF operates using
                  neighborhood status and triggers from a
                  cross-layer information base for dynamic relay
                  set selection and maintenance
           "
       SYNTAX  INTEGER {
                         independent (1),
                         routing (2),
                         crossLayer (3)
                         -- future (4-255)
               }

   SmfRssaID ::= TEXTUAL-CONVENTION
       STATUS       current
       DESCRIPTION
           "An index that identifies through reference to a specific
            RSSA algorithms.  Several are currently defined
            in the appendix of
           "
       SYNTAX      INTEGER {
                            cF(1),
                            sMPR(2),
                            eCDS(3),
                            mprCDS(4)
                            -- future(5-127)
                            -- noStdAction(128-239)
                            -- experimental(240-255)
                   }



   --
   -- Top-Level Object Identifier Assignments
   --

   smfMIBNotifications OBJECT IDENTIFIER ::= { manetSmfMIB 0 }
```

```
   smfMIBObjects       OBJECT IDENTIFIER ::= { manetSmfMIB 1 }
   smfMIBConformance   OBJECT IDENTIFIER ::= { manetSmfMIB 2 }



   --
   -- smfMIBObjects Assignments:
   --      smfCapabilitiesGroup  - 1
   --      smfConfigurationGroup - 2
   --      smfStateGroup         - 3
   --      smfPerformanceGroup   - 4
   --

   --
   -- smfCapabilitiesGroup
   --
   --    This group contains the SMF objects that identify specific
   --    capabilities within this device related to SMF functions.
   --

   smfCapabilitiesGroup  OBJECT IDENTIFIER ::= { smfMIBObjects 1 }

  --
   -- SMF Operational Mode Capabilities Table
   --

   smfOpModeCapabilitiesTable OBJECT-TYPE
       SYNTAX      SEQUENCE OF SmfOpModeCapabilitiesEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "The smfOpModeCapabilitiesTable identifies the
            resident set of SMF Operational Modes on this
            router.
           "
       ::= { smfCapabilitiesGroup 1 }

   smfOpModeCapabilitiesEntry OBJECT-TYPE
       SYNTAX      SmfOpModeCapabilitiesEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "Information about a particular operational
            mode.
           "
       INDEX   { smfOpModeCapabilitiesID }
       ::= { smfOpModeCapabilitiesTable 1 }
```

```
   SmfOpModeCapabilitiesEntry ::= SEQUENCE {
         smfOpModeCapabilitiesID              SmfOpModeID,
         smfOpModeCapabilitiesName            SnmpAdminString,
         smfOpModeCapabilitiesReference       SnmpAdminString
   }

   smfOpModeCapabilitiesID     OBJECT-TYPE
       SYNTAX      SmfOpModeID
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "The index for this entry.  This object identifies
            the particular operational mode for this device.
           "
       ::= { smfOpModeCapabilitiesEntry 1 }

   smfOpModeCapabilitiesName OBJECT-TYPE
       SYNTAX      SnmpAdminString
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "The textual name of this operational
            mode.  Current operational modes include:
            Independent Mode, CDS-aware Routing Mode,
            and Cross-layer Mode.  Others may be defined
            in future revisions of [SMF].
           "
       ::= { smfOpModeCapabilitiesEntry 2 }

   smfOpModeCapabilitiesReference OBJECT-TYPE
       SYNTAX      SnmpAdminString
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object contains a reference to the document that
            defines this operational mode.
           "
       ::= { smfOpModeCapabilitiesEntry 3 }


   --
   -- SMF RSSA Capabilities Table
   --

   smfRssaCapabilitiesTable OBJECT-TYPE
       SYNTAX      SEQUENCE OF SmfRssaCapabilitiesEntry
       MAX-ACCESS  not-accessible
       STATUS      current
```

```
       DESCRIPTION
           "The smfRssaCapabilitiesTable contains
            reference to the specific set of RSSAs
            currently supported on this device.
           "
       ::= { smfCapabilitiesGroup 2 }

   smfRssaCapabilitiesEntry OBJECT-TYPE
       SYNTAX      SmfRssaCapabilitiesEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "Information about a particular RSSA
            algorithm."
       INDEX   { smfRssaCapabilitiesID }
       ::= { smfRssaCapabilitiesTable 1 }

   SmfRssaCapabilitiesEntry ::= SEQUENCE {
         smfRssaCapabilitiesID              SmfRssaID,
         smfRssaCapabilitiesName            SnmpAdminString,
         smfRssaCapabilitiesReference       SnmpAdminString
   }

   smfRssaCapabilitiesID     OBJECT-TYPE
       SYNTAX      SmfRssaID
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "The index for this entry.  This object identifies
            the particular RSSA algorithm in this MIB
            module.  Example RSSAs are found in the
            appendix of [SMF]."
       ::= { smfRssaCapabilitiesEntry 1 }

   smfRssaCapabilitiesName OBJECT-TYPE
       SYNTAX      SnmpAdminString
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "The textual name of this RSSA algorithm.
            Currently defined names are:
                Classical Flooding - cF,
                Source-based MultiPoint
                    Relay - sMPR,
                Essential Connecting Dominating
                    Set - eCDS,
                MultiPoint Relay Connected
                    Dominating Set - mprCDS.
```

```
          "
       ::= { smfRssaCapabilitiesEntry 2 }

   smfRssaCapabilitiesReference OBJECT-TYPE
       SYNTAX      SnmpAdminString
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object contains a published reference
            to the document that defines this algorithm.
           "
       ::= { smfRssaCapabilitiesEntry 3 }




   --
   -- smfConfigurationGroup
   --
   --    This group contains the SMF objects that configure specific
   --    options that determine the overall performance and operation
   --    of the multicast forwarding process for the router device
   --    and its interfaces.
   --

   smfConfigurationGroup  OBJECT IDENTIFIER ::= { smfMIBObjects 2 }

   smfAdminStatus  OBJECT-TYPE
      SYNTAX      SmfStatus
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "The configured status of the SMF process
          on this device.  Enabled(1) means that
          SMF is configured to run on this device.
          Disabled(2) mean that the SMF process
          is configured off.

          This object is persistent and when written
          the entity SHOULD save the change to
          non-volatile storage.
         "
   ::= { smfConfigurationGroup 1 }

   -- Note: need to better define the algorithm to
   --   choose the smfRouterID.
   smfRouterIDAddrType  OBJECT-TYPE
      SYNTAX      InetAddressType
```

```
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "The address type of the address used for
          SMF ID of this router as specified
          in the 'smfRouterID' next.

          This can be set by the management station, must
          the smfRouterID must be a routable address
          assigned to this router.  If the management
          station does not assign this value, then the
          router should choose the highest IP address
          assigned to this router.

          This object is persistent and when written
          the entity SHOULD save the change to
          non-volatile storage.
         "
   ::= { smfConfigurationGroup 2 }

   smfRouterID  OBJECT-TYPE
      SYNTAX      InetAddress
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "The IP address used as the SMF router ID.
          this can be set by the management station.
          If not explicitly set, then the device
          should select a routable IP address
          assigned to this router for use as
          the 'smfRouterID'.

          This object is persistent and when written
          the entity SHOULD save the change to
          non-volatile storage.
         "
   ::= { smfConfigurationGroup 3 }

   smfConfiguredOpMode  OBJECT-TYPE
      SYNTAX      INTEGER {
                          withNHDP(1),
                          cdsAwareRouting(2),
                          crossLayer(3),
                          other(4)
                          }
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
```

```
         "The SMF RSS node operational mode as defined
          in the TEXTUAL CONVENTION for 'SmfOpModeID'
          and in [SMF]..

          The value withNHDP(1) indicates Independent
          Mode of operation.

          The value cdsAwareRouting(2) indicates
          CDS-aware Routing Mode of operation.

          The value crossLayer(3) indicates
          Cross-layer Mode of operation.

          This object is persistent and when written
          the entity SHOULD save the change to
          non-volatile storage.
         "
   ::= { smfConfigurationGroup 4 }


   smfConfiguredRssa  OBJECT-TYPE
      SYNTAX      SmfRssaID
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "The SMF RSS currently operational algorithm
          as defined in the TEXTUAL CONVENTION for
          'SmfRssaID' and in [SMF].

          This object is persistent and when written
          the entity SHOULD save the change to
          non-volatile storage.
         "
   ::= { smfConfigurationGroup 5 }

   smfRssaMember  OBJECT-TYPE
      SYNTAX      INTEGER {
                          potential(1),
                          always(2),
                          never(3)
                          }
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "The RSSA downselects a set of forwarders for
          multicast forwarding.  Sometimes it is useful
          to force an agent to be included or excluded
          from the resulting RSS.  This object is a
```

```
          switch to allow for this behavior.

          The value potential(1) allows the selected
          RSSA to determine if this agent is included
          or excluded from the RSS.

          The value always(1) forces the selected
          RSSA include this agent in the RSS.

          The value never(3) forces the selected
          RSSA to exclude this agent from the RSS.

          This object is persistent and when written
          the entity SHOULD save the change to
          non-volatile storage.
         "
   ::= { smfConfigurationGroup 6 }

   smfIpv4Dpd  OBJECT-TYPE
      SYNTAX      INTEGER {
                          identificationBased(1),
                          hashBased(2)
                          }
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "The current method for IPv4 duplicate packet
          detection.

          The value identificationBased(1)
          indicates that the duplicate packet
          detection relies upon header information
          in the multicast packets to identify
          previously received packets.

          The value 'hashBased(2) indicates that the
          routers duplicate packet detection is based
          upon comparing a hash over the packet fields.

          This object is persistent and when written
          the entity SHOULD save the change to
          non-volatile storage.
         "
   ::= { smfConfigurationGroup 7 }

   smfIpv6Dpd  OBJECT-TYPE
      SYNTAX      INTEGER {
                          identificationBased(1),
```

```
                          hashBased(2)
                          }
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "The current method for IPv6 duplicate packet
          detection.

          The values indicate the type of method used
          for duplicate packet detection as described
          the previous description for the object
          `smfIpv4Dpd'.

          This object is persistent and when written
          the entity SHOULD save the change to
          non-volatile storage.
         "
   ::= { smfConfigurationGroup 8 }

   smfMaxPktLifetime  OBJECT-TYPE
      SYNTAX      Integer32 (0..65535)
      UNITS       "Seconds"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "The estimate of the network packet
          traversal time.

          This object is persistent and when written
          the entity SHOULD save the change to
          non-volatile storage.
         "
      DEFVAL { 60 }
   ::= { smfConfigurationGroup 9 }

   smfDpdMaxMemorySize  OBJECT-TYPE
      SYNTAX      Integer32 (0..65535)
      UNITS       "Kilo-Bytes"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "The locally reserved memory for storage
          of cached DPD records for both IPv4 and
          IPv6 methods.

          This object is persistent and when written
          the entity SHOULD save the change to
          non-volatile storage.
```

```
         "
      DEFVAL { 1024 }
   ::= { smfConfigurationGroup 10 }

   smfDpdEntryMaxLifetime  OBJECT-TYPE
      SYNTAX      Integer32 (0..65525)
      UNITS       "Seconds"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "The maximum lifetime of a cached DPD
          record in the local device storage.

          This object is persistent and when written
          the entity SHOULD save the change to
          non-volatile storage.
         "
      DEFVAL { 600 }
   ::= { smfConfigurationGroup 11 }


   --
   -- Configuration of messages to be included in
   -- NHDP message exchanges in support of SMF
   -- operations.
   --

   -- Note: need to clarify whether this is an option
   --  or is required when the smfOpMode is set
   --  to 'independent'.
   smfNhdpRssaMesgTLVIncluded  OBJECT-TYPE
      SYNTAX      TruthValue
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "Indicates whether the associated NHDP messages
          include the RSSA Message TLV, or not.  This
          is an optional SMF operational setting.
          The value true(1) indicates that this TLV is
          included; the value false(2) indicates that it
          is not included.

          This object is persistent and when written
          the entity SHOULD save the change to
          non-volatile storage.
         "
   ::= { smfConfigurationGroup 12 }
```

```
   smfNhdpRssaAddrBlockTLVIncluded  OBJECT-TYPE
      SYNTAX      TruthValue
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
         "Indicates whether the associated NHDP messages
          include the RSSA Address Block TLV, or not.
          This is an optional SMF operational setting.
          The value true(1) indicates that this TLV is
          included; the value false(2) indicates that it
          is not included.

          This object is persistent and when written
          the entity SHOULD save the change to
          non-volatile storage.
         "
   ::= { smfConfigurationGroup 13 }



   --
   -- Table identifying configured multicast addresses to be forwarded.
   --

   smfConfiguredAddrForwardingTable  OBJECT-TYPE
      SYNTAX     SEQUENCE OF SmfConfiguredAddrForwardingEntry
      MAX-ACCESS not-accessible
      STATUS     current
      DESCRIPTION
         "The (conceptual) table containing information on multicast
          addresses which are to be forwarded by the SMF process.

          Entries in this table are configured.  As well, addresses
          to be forwarded by the SMF device can be dynamically
          discovered by other means.  The corresponding state
          table, smfDiscoveredAddrForwardingTable, contains
          these additional, dynamically discovered address for
          forwarding.

          Each row is associated with a range of multicast
          addresses, and ranges for different rows must be disjoint.

          The objects in this table are persistent and when written
          the entity SHOULD save the change to
          non-volatile storage.
         "
   ::= { smfConfigurationGroup 15 }
```

```
   smfConfiguredAddrForwardingEntry OBJECT-TYPE
      SYNTAX     SmfConfiguredAddrForwardingEntry
      MAX-ACCESS not-accessible
      STATUS     current
      DESCRIPTION
         "An entry (conceptual row) containing the information on a
          particular multicast scope."
      INDEX { smfConfiguredAddrForwardingAddrType,
              smfConfiguredAddrForwardingFirstAddr }
      ::= { smfConfiguredAddrForwardingTable 1 }

   SmfConfiguredAddrForwardingEntry ::= SEQUENCE {
      smfConfiguredAddrForwardingAddrType      InetAddressType,
      smfConfiguredAddrForwardingFirstAddr     InetAddress,
      smfConfiguredAddrForwardingLastAddr      InetAddress,
      smfConfiguredAddrForwardingStatus        RowStatus
   }

   smfConfiguredAddrForwardingAddrType OBJECT-TYPE
      SYNTAX     InetAddressType
      MAX-ACCESS not-accessible
      STATUS     current
      DESCRIPTION
         "The type of the addresses in the multicast forwarding
          range.  Legal values correspond to the subset of
          address families for which multicast address allocation
          is supported."
   ::= { smfConfiguredAddrForwardingEntry 1 }

   smfConfiguredAddrForwardingFirstAddr OBJECT-TYPE
      SYNTAX     InetAddress (SIZE(0..20))
      MAX-ACCESS not-accessible
      STATUS     current
      DESCRIPTION
         "The first address in the multicast scope range.  The type
          of this address is determined by the value of the
          smfConfiguredAddrForwardingAddrType object."
   ::= { smfConfiguredAddrForwardingEntry 2 }

   smfConfiguredAddrForwardingLastAddr OBJECT-TYPE
      SYNTAX     InetAddress (SIZE(0..20))
      MAX-ACCESS read-create
      STATUS     current
      DESCRIPTION
         "The last address in the multicast scope range.
          The type of this address is determined by the
          value of the smfConfiguredAddrForwardingAddrType
          object."
```

```
   ::= { smfConfiguredAddrForwardingEntry 3 }

   smfConfiguredAddrForwardingStatus OBJECT-TYPE
      SYNTAX     RowStatus
      MAX-ACCESS read-create
      STATUS     current
      DESCRIPTION
         "The status of this row, by which new entries may be
          created, or old entries deleted from this table.  If write
          access is supported, the other writable objects in this
          table may be modified even while the status is 'active'."
   ::= { smfConfiguredAddrForwardingEntry 4 }



   --
   -- SMF Interfaces Configuration Table
   --

   smfInterfaceTable  OBJECT-TYPE
      SYNTAX      SEQUENCE OF SmfInterfaceEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "The SMF Interface Table describes the SMF
          interfaces that are participating in the
          SMF packet forwarding process. The ifIndex is
          from the interfaces group defined in the
          Interfaces Group MIB.

          The objects in this table are persistent
          and when written the entity SHOULD save
          the change to non-volatile storage.
          "
      REFERENCE
         "RFC 2863 - The Interfaces Group MIB, McCloghrie,
          K., and F. Kastenholtz, June 2000."
   ::= { smfConfigurationGroup 16 }

   smfInterfaceEntry OBJECT-TYPE
      SYNTAX      SmfInterfaceEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "The SMF interface entry describes one SMF
          interface as indexed by its ifIndex."
      INDEX { smfIfIndex }
   ::= { smfInterfaceTable 1 }
```

```
   SmfInterfaceEntry ::=
      SEQUENCE {
         smfIfIndex        InterfaceIndexOrZero,
         smfIfAdminStatus  SmfStatus,
         smfIfRowStatus    RowStatus
         }

   smfIfIndex  OBJECT-TYPE
      SYNTAX      InterfaceIndexOrZero
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The ifIndex for this SMF interface."
      ::= { smfInterfaceEntry 1 }

   smfIfAdminStatus OBJECT-TYPE
      SYNTAX      SmfStatus
      MAX-ACCESS  read-create
      STATUS      current
      DESCRIPTION
          "The SMF interface's administrative status.
          The value 'enabled' denotes that the interface
          is running the SMF forwarding process.
          The value 'disabled' denotes that the interface is
          external to the SMF forwarding process.
          "
      ::= { smfInterfaceEntry 2 }

   smfIfRowStatus  OBJECT-TYPE
      SYNTAX      RowStatus
      MAX-ACCESS  read-create
      STATUS      current
      DESCRIPTION
         "This object permits management of the table
          by facilitating actions such as row creation,
          construction, and destruction. The value of
          this object has no effect on whether other
          objects in this conceptual row can be
          modified."
   ::= { smfInterfaceEntry 3 }



   --
   -- smfStateGroup
   --
   --    Contains information describing the current state of the SMF
   --    process such as the current inclusion in the RS or not.
```

```
   --

   smfStateGroup  OBJECT IDENTIFIER ::= { smfMIBObjects 3 }

   smfNodeRsStatusIncluded  OBJECT-TYPE
      SYNTAX      TruthValue
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The current status of the SMF node in the context of
          the MANETs relay set. A value of true(1) indicates
          that the node is currently part of the MANET Relay
          Set. A value of false(2) indicates that the node
          is currently not part of the MANET Relay Set."
   ::= { smfStateGroup 1 }

   smfDpdMemoryOverflow  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The number of times that the memory for caching
          records for DPD overran and records had to be flushed.
          The number of records to be flushed upon a buffer
          overflow is an implementation specific decision.
         "
   ::= { smfStateGroup 2 }



   --
   -- Dynamically Discovered Multicast Addr Table
   --

   smfDiscoveredAddrForwardingTable  OBJECT-TYPE
      SYNTAX     SEQUENCE OF SmfDiscoveredAddrForwardingEntry
      MAX-ACCESS not-accessible
      STATUS     current
      DESCRIPTION
         "The (conceptual) table containing information on multicast
          addresses which are to be forwarded by the SMF process.

          Entries in this table are configured.  As well, addresses
          to be forwarded by the SMF device can be dynamically
          discovered by other means.  The corresponding state
          table, smfDiscoveredAddrForwardingTable contains
          these additional, dynamically discovered address for
          forwarding.
```

```
          Each row is associated with a range of
          multicast addresses, and ranges for different rows
          must be disjoint.
         "
   ::= { smfStateGroup 3 }

   smfDiscoveredAddrForwardingEntry OBJECT-TYPE
      SYNTAX     SmfDiscoveredAddrForwardingEntry
      MAX-ACCESS not-accessible
      STATUS     current
      DESCRIPTION
         "An entry (conceptual row) containing the information on a
          particular multicast scope."
      INDEX { smfDiscoveredAddrForwardingAddrType,
              smfDiscoveredAddrForwardingFirstAddr }
      ::= { smfDiscoveredAddrForwardingTable 1 }

   SmfDiscoveredAddrForwardingEntry ::= SEQUENCE {
      smfDiscoveredAddrForwardingAddrType   InetAddressType,
      smfDiscoveredAddrForwardingFirstAddr  InetAddress,
      smfDiscoveredAddrForwardingLastAddr   InetAddress,
      smfDiscoveredAddrForwardingStatus     RowStatus
   }

   smfDiscoveredAddrForwardingAddrType OBJECT-TYPE
      SYNTAX     InetAddressType
      MAX-ACCESS not-accessible
      STATUS     current
      DESCRIPTION
         "The type of the addresses in the multicast forwarding
          range.  Legal values correspond to the subset of
          address families for which multicast address allocation
          is supported."
   ::= { smfDiscoveredAddrForwardingEntry 1 }

   smfDiscoveredAddrForwardingFirstAddr OBJECT-TYPE
      SYNTAX     InetAddress (SIZE(0..20))
      MAX-ACCESS not-accessible
      STATUS     current
      DESCRIPTION
         "The first address in the multicast scope range.  The type
          of this address is determined by the value of the
          smfConfiguredAddrForwardingAddrType object."
   ::= { smfDiscoveredAddrForwardingEntry 2 }

   smfDiscoveredAddrForwardingLastAddr OBJECT-TYPE
      SYNTAX     InetAddress (SIZE(0..20))
      MAX-ACCESS read-create
```

```
      STATUS     current
      DESCRIPTION
         "The last address in the multicast scope range.
          The type of this address is determined by the
          value of the smfConfiguredAddrForwardingAddrType
          object."
   ::= { smfDiscoveredAddrForwardingEntry 3 }

   smfDiscoveredAddrForwardingStatus OBJECT-TYPE
      SYNTAX     RowStatus
      MAX-ACCESS read-create
      STATUS     current
      DESCRIPTION
         "The status of this row, by which new entries may be
          created, or old entries deleted from this table.  If write
          access is supported, the other writable objects in this
          table may be modified even while the status is `active'."
   ::= { smfDiscoveredAddrForwardingEntry 4 }



   --
   -- SMF Neighbor Table
   --

   smfNeighborTable  OBJECT-TYPE
      SYNTAX       SEQUENCE OF SmfNeighborEntry
      MAX-ACCESS   not-accessible
      STATUS       current
      DESCRIPTION
         "The SMF NeighborTable describes the
          current neighbor nodes, their address
          and SMF RSSA and the interface on which
          they can be reached."
      REFERENCE
         "Simplified Multicast Forwarding for MANET
          (SMF), Macker, J., July 2009.
          Section 7: SMF Neighborhood Discovery
          Requirements."
   ::= { smfStateGroup 4 }

   smfNeighborEntry  OBJECT-TYPE
      SYNTAX       SmfNeighborEntry
      MAX-ACCESS   not-accessible
      STATUS       current
      DESCRIPTION
         "The SMF Neighbor Table contains the
          set of one-hop neighbors, the interface
```

```
          they are reachable on and the SMF RSSA
          they are currently running."
      INDEX { smfNeighborIpAddrType,
              smfNeighborIpAddr,
              smfNeighborPrefixLen }
   ::= { smfNeighborTable 1 }

   SmfNeighborEntry ::=
      SEQUENCE {
         smfNeighborIpAddrType        InetAddressType,
         smfNeighborIpAddr            InetAddress,
         smfNeighborPrefixLen         InetAddressPrefixLength,
         smfNeighborRSSA              SmfRssaID,
         smfNeighborNextHopInterface  InterfaceIndexOrZero
         }

   smfNeighborIpAddrType  OBJECT-TYPE
      SYNTAX      InetAddressType
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "The neighbor IP address type."
   ::= { smfNeighborEntry 1 }

   smfNeighborIpAddr  OBJECT-TYPE
      SYNTAX      InetAddress
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "The neighbor Inet IPv4 or IPv6 address."
   ::= { smfNeighborEntry 2 }

   smfNeighborPrefixLen  OBJECT-TYPE
      SYNTAX      InetAddressPrefixLength
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "The prefix length. This is a decimal value that
          indicates the number of contiguous, higher-order
          bits of the address that make up the network
          portion of the address."
   ::= { smfNeighborEntry 3 }

   smfNeighborRSSA  OBJECT-TYPE
      SYNTAX       SmfRssaID
      MAX-ACCESS   read-only
      STATUS       current
      DESCRIPTION
```

```
         "The current RSSA running on the neighbor.
          The list is identical to that described
          above for the smfRssa object."
   ::= { smfNeighborEntry 4 }

   smfNeighborNextHopInterface OBJECT-TYPE
      SYNTAX       InterfaceIndexOrZero
      MAX-ACCESS   read-only
      STATUS       current
      DESCRIPTION
         "The interface ifIndex over which the
          neighbor is reachable in one-hop."
   ::= { smfNeighborEntry 5 }




   --
   -- SMF Performance Group
   --
   --    Contains objects which help to characterize the
   --    performance of the SMF RSSA process, such as statistics
   --    counters. There are two types of SMF RSSA statistics:
   --    global counters and per interface counters.
   --

   smfPerformanceGroup  OBJECT IDENTIFIER ::= { smfMIBObjects 4 }

   smfGlobalPerfGroup  OBJECT IDENTIFIER ::= { smfPerformanceGroup 1 }

   --
   -- IPv4 packet counters
   --

   smfIpv4MultiPktsRecvTotal  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter of the total number of
          multicast IPv4 packets received by the
          device."
   ::= { smfGlobalPerfGroup 1 }

   smfIpv4MultiPktsForwardedTotal  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
```

```
      DESCRIPTION
         "A counter of the total number of
          multicast IPv4 packets forwarded by the
          device."
   ::= { smfGlobalPerfGroup 2 }

   smfIpv4DuplMultiPktsDetectedTotal  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter of the total number of duplicate
          multicast IPv4 packets detected by the
          device."
   ::= { smfGlobalPerfGroup 3 }

   smfIpv4DroppedMultiPktsTTLExceededTotal  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter of the total number of dropped
          multicast IPv4 packets by the
          device due to TTL exceeded."
   ::= { smfGlobalPerfGroup 4 }

   smfIpv4TTLLargerThanPreviousTotal  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter of the total number of IPv4 packets
          recieved which have a TTL larger than that
          of a previously received identical packet.
         "
   ::= { smfGlobalPerfGroup 5 }

   --
   -- IPv6 packet counters
   --

   smfIpv6MultiPktsRecvTotal  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter of the total number of
          multicast IPv6 packets received by the
```

```
          device."
   ::= { smfGlobalPerfGroup 6 }

   smfIpv6MultiPktsForwardedTotal  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter of the total number of
          multicast IPv6 packets forwarded by the
          device."
   ::= { smfGlobalPerfGroup 7 }

   smfIpv6DuplMultiPktsDetectedTotal  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter of the total number of duplicate
          multicast IPv6 packets detected by the
          device."
   ::= { smfGlobalPerfGroup 8 }

   smfIpv6DroppedMultiPktsTTLExceededTotal  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter of the total number of dropped
          multicast IPv6 packets by the
          device due to TTL exceeded."
   ::= { smfGlobalPerfGroup 9 }

   smfIpv6TTLLargerThanPreviousTotal  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter of the total number of IPv6 packets
          recieved which have a TTL larger than that
          of a previously recived identical packet.
         "
   ::= { smfGlobalPerfGroup 10 }

   smfIpv6HAVAssistsReqdTotal  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
```

```
      DESCRIPTION
         "A counter of the total number of IPv6 packets
          recieved which required the HAV assist for DPD.
         "
   ::= { smfGlobalPerfGroup 11 }

   smfIpv6DpdHeaderInsertionsTotal  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter of the total number of IPv6 packets
          recieved which the device inserted the
          DPD header option.
         "
   ::= { smfGlobalPerfGroup 12 }


   --
   -- Per SMF Interface Performance Table
   --

   smfInterfacePerfGroup OBJECT IDENTIFIER ::= { smfPerformanceGroup 2 }

   smfIpv4InterfacePerfTable OBJECT-TYPE
      SYNTAX       SEQUENCE OF SmfIpv4InterfacePerfEntry
      MAX-ACCESS   not-accessible
      STATUS       current
      DESCRIPTION
         "The SMF Interface Performance Table
          describes the SMF statistics per
          interface."
   ::= { smfInterfacePerfGroup 1 }

   smfIpv4InterfacePerfEntry OBJECT-TYPE
      SYNTAX       SmfIpv4InterfacePerfEntry
      MAX-ACCESS   not-accessible
      STATUS       current
      DESCRIPTION
         "The SMF Interface Performance entry
          describes the statistics for a particular
          node interface."
      INDEX { smfIpv4IfPerfIfIndex }
   ::= { smfIpv4InterfacePerfTable 1 }

   SmfIpv4InterfacePerfEntry ::=
      SEQUENCE {
         smfIpv4IfPerfIfIndex                    InterfaceIndexOrZero,
```

```
         smfIpv4MultiPktsRecvPerIf                Counter32,
         smfIpv4MultiPktsForwardedPerIf           Counter32,
         smfIpv4DuplMultiPktsDetectedPerIf        Counter32,
         smfIpv4DroppedMultiPktsTTLExceededPerIf Counter32,
         smfIpv4TTLLargerThanPreviousPerIf        Counter32
         }

   smfIpv4IfPerfIfIndex  OBJECT-TYPE
      SYNTAX      InterfaceIndexOrZero
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "The ifIndex for this node interface
          that is collecting this set of
          performance management statistics."
   ::= { smfIpv4InterfacePerfEntry 1 }

   smfIpv4MultiPktsRecvPerIf  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter of the number of
          multicast IP packets received by the
          device on this interface."
   ::= { smfIpv4InterfacePerfEntry 2 }

   smfIpv4MultiPktsForwardedPerIf  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter of the number of
          multicast IP packets forwarded by the
          device on this interface."
   ::= { smfIpv4InterfacePerfEntry 3 }

   smfIpv4DuplMultiPktsDetectedPerIf  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter of the number of duplicate
          multicast IP packets detected by the
          device on this interface."
   ::= { smfIpv4InterfacePerfEntry 4 }

   smfIpv4DroppedMultiPktsTTLExceededPerIf  OBJECT-TYPE
```

```
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter of the total number of dropped
          multicast IPv4 packets by the
          device due to TTL exceeded."
   ::= { smfIpv4InterfacePerfEntry 5 }

   smfIpv4TTLLargerThanPreviousPerIf  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter of the total number of IPv4 packets
          recieved which have a TTL larger than that
          of a previously recived identical packet.
         "
   ::= { smfIpv4InterfacePerfEntry 6 }


   smfIpv6InterfacePerfTable OBJECT-TYPE
      SYNTAX       SEQUENCE OF SmfIpv6InterfacePerfEntry
      MAX-ACCESS   not-accessible
      STATUS       current
      DESCRIPTION
         "The SMF Interface Performance Table
          describes the SMF statistics per
          interface."
   ::= { smfInterfacePerfGroup 2 }

   smfIpv6InterfacePerfEntry OBJECT-TYPE
      SYNTAX       SmfIpv6InterfacePerfEntry
      MAX-ACCESS   not-accessible
      STATUS       current
      DESCRIPTION
         "The SMF Interface Performance entry
          describes the statistics for a particular
          node interface."
      INDEX { smfIpv6IfPerfIfIndex }
   ::= { smfIpv6InterfacePerfTable 1 }

   SmfIpv6InterfacePerfEntry ::=
      SEQUENCE {
         smfIpv6IfPerfIfIndex                  InterfaceIndexOrZero,
         smfIpv6MultiPktsRecvPerIf             Counter32,
         smfIpv6MultiPktsForwardedPerIf        Counter32,
         smfIpv6DuplMultiPktsDetectedPerIf     Counter32,
```

```
         smfIpv6DroppedMultiPktsTTLExceededPerIf Counter32,
         smfIpv6TTLLargerThanPreviousPerIf       Counter32,
         smfIpv6HAVAssistsReqdPerIf              Counter32,
         smfIpv6DpdHeaderInsertionsPerIf         Counter32
         }

   smfIpv6IfPerfIfIndex  OBJECT-TYPE
      SYNTAX      InterfaceIndexOrZero
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "The ifIndex for this node interface
          that is collecting this set of
          performance management statistics.

          For packets generated locally at
          this node, performance counters
          are assigned to the loopback
          interface.
         "
   ::= { smfIpv6InterfacePerfEntry 1 }

   smfIpv6MultiPktsRecvPerIf  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter of the number of
          multicast IP packets received by the
          device on this interface."
   ::= { smfIpv6InterfacePerfEntry 2 }

   smfIpv6MultiPktsForwardedPerIf  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter of the number of
          multicast IP packets forwarded by the
          device on this interface."
   ::= { smfIpv6InterfacePerfEntry 3 }

   smfIpv6DuplMultiPktsDetectedPerIf  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter of the number of duplicate
```

```
          multicast IP packets detected by the
          device on this interface."
   ::= { smfIpv6InterfacePerfEntry 4 }

   smfIpv6DroppedMultiPktsTTLExceededPerIf  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter of the number of dropped
          multicast IP packets by the
          device on this interface due to TTL
          exceeded."
   ::= { smfIpv6InterfacePerfEntry 5 }

   smfIpv6TTLLargerThanPreviousPerIf  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter of the total number of IPv6 packets
          recieved which have a TTL larger than that
          of a previously recived identical packet.
         "
   ::= { smfIpv6InterfacePerfEntry 6 }

   smfIpv6HAVAssistsReqdPerIf  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter of the total number of IPv6 packets
          recieved which required the HAV assist for DPD.
         "
   ::= { smfIpv6InterfacePerfEntry 7 }

   smfIpv6DpdHeaderInsertionsPerIf  OBJECT-TYPE
      SYNTAX      Counter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "A counter of the total number of IPv6 packets
          recieved which the device inserted the
          DPD header option.
         "
   ::= { smfIpv6InterfacePerfEntry 8 }
```

```
   --
   -- Notifications
   --

smfMIBNotifControl OBJECT IDENTIFIER ::= { smfMIBNotifications 1 }
smfMIBNotifObjects OBJECT IDENTIFIER ::= { smfMIBNotifications 2 }
smfMIBNotifStates  OBJECT IDENTIFIER ::= { smfMIBNotifications 3 }


   -- smfMIBNotifControl
   smfSetNotification OBJECT-TYPE
          SYNTAX       OCTET STRING (SIZE(4))
          MAX-ACCESS   read-write
          STATUS       current
          DESCRIPTION
             "A 4-octet string serving as a bit map for
             the notification events defined by the SMF MIB
             notifications. This object is used to enable
             and disable specific SMF MIB notifications where
             a 1 in the bit field represents enabled. The
             right-most bit (least significant) represents
             notification 0.

             This object is persistent and when written
             the entity SHOULD save the change to
             non-volatile storage.
             "
           ::= { smfMIBNotifControl 1 }

   smfDpdMemoryOverflowThreshold OBJECT-TYPE
          SYNTAX       Integer32 (0..255)
          MAX-ACCESS   read-write
          STATUS       current
          DESCRIPTION
             "A threshold value for the
              `smfDpdmemoryOverflowEvents' object.
              If the number of occurences exceeds
              this threshold within the previous
              number of seconds
              'smfDpdMemoryOverflowWindow',
              then the `smfDpdMemoryOverflowEvent'
              notification is sent.
             "
           ::= { smfMIBNotifControl 2 }

   smfDpdMemoryOverflowWindow OBJECT-TYPE
          SYNTAX       TimeTicks
          MAX-ACCESS   read-write
```

```
          STATUS       current
          DESCRIPTION
             "A time window value for the
              'smfDpdmemoryOverflowEvents' object.
              If the number of occurences exceeds
              the 'smfDpdMemoryOverflowThreshold'
              within the previous number of seconds
              'smfDpdMemoryOverflowWindow',
              then the 'smfDpdMemoryOverflowEvent'
              notification is sent.
             "
           ::= { smfMIBNotifControl 3 }

   smfIpv4DuplMultiPktsDetectedTotalThreshold OBJECT-TYPE
          SYNTAX       Integer32 (0..255)
          MAX-ACCESS   read-write
          STATUS       current
          DESCRIPTION
             "A threshold value for the
              'smfIpv4DuplMultiPktsDetectedTotal'
              object.  If the number of occurences
              exceeds this threshold within the
              previous number of seconds
              'smfIpv4DuplMultiPktsDetectedTotalWindow',
              then the
              'smfIpv4DuplMultiPktsDetectedTotalEvent'
              notification is sent.
             "
           ::= { smfMIBNotifControl 4 }

   smfIpv4DuplMultiPktsDetectedTotalWindow OBJECT-TYPE
          SYNTAX       TimeTicks
          MAX-ACCESS   read-write
          STATUS       current
          DESCRIPTION
             "A time window value for the
              'smfIpv4DuplMultiPktsDetectedTotalEvents'
              object.  If the number of occurences
              exceeds the
              'smfIpv4DuplMultiPktsDetectedTotalThreshold'
              within the previous number of seconds
              'smfIpv4DuplMultiPktsDetectedTotalWindow',
              then the
              'smfIpv4DuplMultiPktsDetectedTotalEvent'
              notification is sent.
             "
           ::= { smfMIBNotifControl 5 }
```

```
   smfIpv6DuplMultiPktsDetectedTotalThreshold OBJECT-TYPE
          SYNTAX       Integer32 (0..255)
          MAX-ACCESS   read-write
          STATUS       current
          DESCRIPTION
             "A threshold value for the
              `smfIpv6DuplMultiPktsDetectedTotal'
              object.  If the number of occurences
              exceeds this threshold within the
              previous number of seconds
              `smfIpv6DuplMultiPktsDetectedTotalWindow',
              then the
              `smfIpv6DuplMultiPktsDetectedTotalEvent'
              notification is sent.
             "
           ::= { smfMIBNotifControl 6 }

   smfIpv6DuplMultiPktsDetectedTotalWindow OBJECT-TYPE
          SYNTAX       TimeTicks
          MAX-ACCESS   read-write
          STATUS       current
          DESCRIPTION
             "A time window value for the
              `smfIpv6DuplMultiPktsDetectedTotalEvents'
              object.  If the number of occurences
              exceeds the
              `smfIpv6DuplMultiPktsDetectedTotalThreshold'
              within the previous number of seconds
              'smfIpv6DuplMultiPktsDetectedTotalWindow',
              then the
              `smfIpv6DuplMultiPktsDetectedTotalEvent'
              notification is sent.
             "
           ::= { smfMIBNotifControl 7 }



   -- smfMIBNotifObjects

   smfAdminStatusChange NOTIFICATION-TYPE
          OBJECTS { smfRouterIDAddrType, -- The originator of
                                         --     the notification.
                    smfRouterID,         -- The originator of
                                         --     the notification.
                    smfAdminStatus       -- The new status of the
                                         --     SMF process.
                  }
          STATUS       current
```

```
          DESCRIPTION
             "smfAdminStatusChange is a notification sent when a
              the 'smfAdminStatus' object changes.
             "
          ::= { smfMIBNotifObjects 1 }

   smfConfiguredOpModeChange NOTIFICATION-TYPE
          OBJECTS { smfRouterIDAddrType, -- The originator of
                                         --     the notification.
                    smfRouterID,         -- The originator of
                                         --     the notification.
                    smfConfiguredOpMode  -- The new Operations
                                         --     Mode of the SMF
                                         --     process.
                  }
          STATUS       current
          DESCRIPTION
             "smfConfiguredOpModeChange is a notification
              sent when a the 'smfConfiguredOpMode' object
              changes.
             "
          ::= { smfMIBNotifObjects 2 }

   smfConfiguredRssaChange NOTIFICATION-TYPE
          OBJECTS { smfRouterIDAddrType, -- The originator of
                                         --     the notification.
                    smfRouterID,         -- The originator of
                                         --     the notification.
                    smfConfiguredRssa -- The new RSSA for
                                         --     the SMF process.
                  }
          STATUS       current
          DESCRIPTION
             "smfAdminStatusChange is a notification sent when a
              the 'smfConfiguredRssa' object changes.
             "
          ::= { smfMIBNotifObjects 3 }

   smfIfAdminStatusChange NOTIFICATION-TYPE
          OBJECTS { smfRouterIDAddrType, -- The originator of
                                         --     the notification.
                    smfRouterID,          -- The originator of
                                         --     the notification.
                    smfIfIndex,          -- The interface whose
                                         --     status has changed.
                    smfIfAdminStatus   -- The new status of the
                                         --     SMF interface.
                  }
```

```
        STATUS       current
        DESCRIPTION
           "smfIfAdminStatusChange is a notification sent when a
            the 'smfIfAdminStatus' object changes.
           "
        ::= { smfMIBNotifObjects 4 }

    smfDpdMemoryOverflowEvent NOTIFICATION-TYPE
        OBJECTS { smfRouterIDAddrType, -- The originator of
                                       --     the notification.
                  smfRouterID,         -- The originator of
                                       --     the notification.
                  smfDpdMemoryOverflow -- The counter of
                                       --     the overflows.
           }
        STATUS       current
        DESCRIPTION
           "smfDpdMemoryOverflowEvents is sent when the
            number of memory overflow events exceeds the
            the 'smfDpdMemoryOverflowThreshold' within the
            previous number of seconds defined by the
            'smfDpdMemoryOverflowWindow'.
           "
        ::= { smfMIBNotifObjects 5 }

   smfIpv4DuplMultiPktsDetectedTotalEvents NOTIFICATION-TYPE
        OBJECTS { smfRouterIDAddrType, -- The originator of
                                       --     the notification.
                  smfRouterID,         -- The originator of
                                       --     the notification.
                  smfIpv4DuplMultiPktsDetectedTotal -- The
                                       --     counter of detected
                                       --     duplicates.
           }
        STATUS       current
        DESCRIPTION
           "smfIpv4DuplMultiPktsDetectedTotal is a
            notification sent when the number of
            IPv4 duplicate packets detected exceeds the
            'smfIpv4DuplMultiPktsDetectedTotalThreshold'
            during the previous number of seconds
            'smfIpv4DuplPktsDetectedTotalWindow'.
           "
        ::= { smfMIBNotifObjects 6 }

   smfIpv6DuplMultiPktsDetectedTotalEvents NOTIFICATION-TYPE
        OBJECTS { smfRouterIDAddrType, -- The originator of
                                       --     the notification.
```

```
                    smfRouterID,      -- The originator of
                                      --     the notification.
                    smfIpv6DuplMultiPktsDetectedTotal -- The
                                      --     counter of detected
                                      --     duplicates.
             }
          STATUS       current
          DESCRIPTION
             "smfIpv6DuplMultiPktsDetectedTotal is a
              notification sent when the number of
              IPv6 duplicate packets detected exceeds the
              'smfIpv6DuplMultiPktsDetectedTotalThreshold'
              during the previous number of seconds
              'smfIpv6DuplPktsDetectedTotalWindow'.
             "
          ::= { smfMIBNotifObjects 7 }



    -- smfMIBNotifStates
    --   is empty.



   --
   -- Compliance Statements
   --

   smfCompliances  OBJECT IDENTIFIER ::= { smfMIBConformance 1 }
   smfMIBGroups    OBJECT IDENTIFIER ::= { smfMIBConformance 2 }

   smfBasicCompliance  MODULE-COMPLIANCE
      STATUS current
      DESCRIPTION "The basic implementation requirements for
                   managed network entities that implement
                   the SMF RSSA process."
      MODULE  -- this module
      MANDATORY-GROUPS { smfCapabObjectsGroup,
                         smfConfigObjectsGroup }
   ::= { smfCompliances 1 }

   smfFullCompliance MODULE-COMPLIANCE
      STATUS current
      DESCRIPTION "The full implementation requirements for
                   managed network entities that implement
                   the SMF RSSA process."
```

```
      MODULE  -- this module
      MANDATORY-GROUPS { smfCapabObjectsGroup,
                         smfConfigObjectsGroup,
                         smfStateObjectsGroup,
                         smfPerfObjectsGroup,
                         smfNotifObjectsGroup,
                         smfNotificationsGroup
                       }
   ::= { smfCompliances 2 }

   --
   -- Units of Conformance
   --

   smfCapabObjectsGroup OBJECT-GROUP
      OBJECTS {
              smfOpModeCapabilitiesName,
              smfOpModeCapabilitiesReference,

              smfRssaCapabilitiesName,
              smfRssaCapabilitiesReference
      }
      STATUS  current
      DESCRIPTION
         "Set of SMF configuration objects implemented
          in this module."
   ::= { smfMIBGroups 1 }

   smfConfigObjectsGroup OBJECT-GROUP
      OBJECTS {
              smfAdminStatus,
              smfRouterIDAddrType,
              smfRouterID,
              smfIfIndex,
              smfConfiguredOpMode,
              smfConfiguredRssa,
              smfRssaMember,
              smfIpv4Dpd,
              smfIpv6Dpd,
              smfMaxPktLifetime,
              smfDpdMaxMemorySize,
              smfDpdEntryMaxLifetime,
              smfNhdpRssaMesgTLVIncluded,
              smfNhdpRssaAddrBlockTLVIncluded,

              smfConfiguredAddrForwardingLastAddr,
              smfConfiguredAddrForwardingStatus,
```

```
              smfIfAdminStatus,
              smfIfRowStatus
      }
      STATUS  current
      DESCRIPTION
         "Set of SMF configuration objects implemented
          in this module."
   ::= { smfMIBGroups 2 }

   smfStateObjectsGroup  OBJECT-GROUP
      OBJECTS {
              smfNodeRsStatusIncluded,
              smfDpdMemoryOverflow,

              smfDiscoveredAddrForwardingLastAddr,
              smfDiscoveredAddrForwardingStatus,

              smfNeighborRSSA,
              smfNeighborNextHopInterface
      }
      STATUS  current
      DESCRIPTION
         "Set of SMF state objects implemented
          in this module."
   ::= { smfMIBGroups 3 }

   smfPerfObjectsGroup  OBJECT-GROUP
      OBJECTS {
              smfIpv4MultiPktsRecvTotal,
              smfIpv4MultiPktsForwardedTotal,
              smfIpv4DuplMultiPktsDetectedTotal,
              smfIpv4DroppedMultiPktsTTLExceededTotal,
              smfIpv4TTLLargerThanPreviousTotal,

              smfIpv6MultiPktsRecvTotal,
              smfIpv6MultiPktsForwardedTotal,
              smfIpv6DuplMultiPktsDetectedTotal,
              smfIpv6DroppedMultiPktsTTLExceededTotal,
              smfIpv6TTLLargerThanPreviousTotal,
              smfIpv6HAVAssistsReqdTotal,
              smfIpv6DpdHeaderInsertionsTotal,

              smfIpv4MultiPktsRecvPerIf,
              smfIpv4MultiPktsForwardedPerIf,
              smfIpv4DuplMultiPktsDetectedPerIf,
              smfIpv4DroppedMultiPktsTTLExceededPerIf,
              smfIpv4TTLLargerThanPreviousPerIf,
```

```
              smfIpv6MultiPktsRecvPerIf,
              smfIpv6MultiPktsForwardedPerIf,
              smfIpv6DuplMultiPktsDetectedPerIf,
              smfIpv6DroppedMultiPktsTTLExceededPerIf,
              smfIpv6TTLLargerThanPreviousPerIf,
              smfIpv6HAVAssistsReqdPerIf,
              smfIpv6DpdHeaderInsertionsPerIf
      }
      STATUS  current
      DESCRIPTION
         "Set of SMF performance objects implemented
          in this module by total and per interface."
   ::= { smfMIBGroups 4 }

   smfNotifObjectsGroup  OBJECT-GROUP
      OBJECTS {
              smfSetNotification,
              smfDpdMemoryOverflowThreshold,
              smfDpdMemoryOverflowWindow,
              smfIpv4DuplMultiPktsDetectedTotalThreshold,
              smfIpv4DuplMultiPktsDetectedTotalWindow,
              smfIpv6DuplMultiPktsDetectedTotalThreshold,
              smfIpv6DuplMultiPktsDetectedTotalWindow
      }
      STATUS  current
      DESCRIPTION
         "Set of SMF notification control
          objects implemented in this module."
   ::= { smfMIBGroups 5 }

   smfNotificationsGroup  NOTIFICATION-GROUP
      NOTIFICATIONS {
              smfAdminStatusChange,
              smfConfiguredOpModeChange,
              smfConfiguredRssaChange,
              smfIfAdminStatusChange,
              smfDpdMemoryOverflowEvent,
              smfIpv4DuplMultiPktsDetectedTotalEvents,
              smfIpv6DuplMultiPktsDetectedTotalEvents
      }
      STATUS  current
      DESCRIPTION
         "Set of SMF notifications implemented
          in this module."
   ::= { smfMIBGroups 6 }
```

END

8. Security Considerations

[TODO] Each specification that defines one or more MIB modules MUST contain a section that discusses security considerations relevant to those modules. This section MUST be patterned after the latest approved template (available at http://www.ops.ietf.org/mib-security.html). Remember that the objective is not to blindly copy text from the template, but rather to think and evaluate the risks/vulnerabilities and then state/ document the result of this evaluation.

[TODO] if you have any read-write and/or read-create objects, please include the following boilerplate paragraph.

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 can have a negative effect on network operations. These are the tables and objects and their sensitivity/vulnerability:

- o [TODO] writable MIB objects that could be especially disruptive if abused MUST be explicitly listed by name and the associated security risks MUST be spelled out; RFC 2669 has a very good example.

- o [TODO] list the writable tables and objects and state why they are sensitive.

[TODO] else if there are no read-write objects in your MIB module, use the following boilerplate paragraph.

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.

[TODO] if you have any sensitive readable objects, please include the following boilerplate paragraph.

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:

o [TODO] you must explicitly list by name any readable objects that are sensitive or vulnerable and the associated security risks MUST be spelled out (for instance, if they might reveal customer information or violate personal privacy laws such as those of the European Union if exposed to unauthorized parties)

o [TODO] list the tables and objects and state why they are sensitive.

[TODO] discuss what security the protocol used to carry the information should have. The following three boilerplate paragraphs should not be changed without very good reason. Changes will almost certainly require justification during IESG review.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPSec), even then, there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

It is RECOMMENDED that implementers consider the security features as provided by the SNMPv3 framework (see [RFC3410], section 8), including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

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.

9. IANA Considerations

[TODO] In order to comply with IESG policy as set forth in http://www.ietf.org/ID-Checklist.html, every Internet-Draft that is submitted to the IESG for publication MUST contain an IANA Considerations section. The requirements for this section vary depending what actions are required of the IANA. see RFC4181 section 3.5 for more information on writing an IANA clause for a MIB module document.

[TODO] select an option and provide the necessary details.

Option #1:

The MIB module in this document uses the following IANA-assigned OBJECT IDENTIFIER values recorded in the SMI Numbers registry:

```
Descriptor        OBJECT IDENTIFIER value
----------        -----------------------

sampleMIB  { mib-2 XXX }
```

Option #2:

Editor's Note (to be removed prior to publication): the IANA is requested to assign a value for "XXX" under the 'mib-2' subtree and to record the assignment in the SMI Numbers registry. When the assignment has been made, the RFC Editor is asked to replace "XXX" (here and in the MIB module) with the assigned value and to remove this note.

Note well: prior to official assignment by the IANA, a draft document MUST use placeholders (such as "XXX" above) rather than actual numbers. See RFC4181 Section 4.5 for an example of how this is done in a draft MIB module.

Option #3:

This memo includes no request to IANA.

10. Contributors

This MIB document uses the template authored by D. Harrington which is based on contributions from the MIB Doctors, especially Juergen Schoenwaelder, Dave Perkins, C.M.Heard and Randy Presuhn.

11. Acknowledgements

12. References

12.1. Normative References

[RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.

[RFC3411] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing Simple Network

Management Protocol (SNMP) Management Frameworks", STD 62, RFC 3411, December 2002.

[RFC3418] Presuhn, R., "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3418, December 2002.

[RFC4001] Daniele, M., Haberman, B., Routhier, S., and J. Schoenwaelder, "Textual Conventions for Internet Network Addresses", RFC 4001, February 2005.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

[I-D.ietf-manet-smf] Macker, J. and S. Team, "Simplified Multicast Forwarding", draft-ietf-manet-smf-10 (work in progress), March 2010.

12.2. Informative References

[RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

Appendix A. Change Log

This section tracks the revision history in the development of this SMF-MIB. It will be removed from the final version of this document.

These changes were made from draft-ietf-manet-smf-mib-01 to draft-ietf-manet-smf-mib-02.

1. Added the NotificationGroup to the MIB and updated the ConformanceGroup.

2. Added the definition of an smfRouterID to the MIB. This is later used in the Notifications to indicate the origin of the event to the management station.

3. Removed the Router Priority object as this was used only in the eCDS algorithm and hence should be contained within the future eCDS-MIB.

4. Cleaned up the TEXTUAL CONVENTION for the 'SmfOpMode'.

5. Filled in some of the missing text in various object descriptions.

These changes were made from draft-ietf-manet-smf-mib-00 to draft-ietf-manet-dsmf-mib-01.

1. Editorial changes to the textual material. These included the addition of the paragraphs on TEXTUAL-CONVENTIONS defined and imported into this MIB and relationships to other MIBs.

2. Identified those objects in the SMF-MIB requiring non-volatile storage.

3. Changed the name of the TEXTUAL-CONVENTION 'Status', defined within this MIB to 'SmfStatus'.

Appendix B. Open Issues

This section contains the set of open issues related to the development and design of the SMF-MIB. This section will not be present in the final version of the MIB and will be removed once all the open issues have been resolved.

1. The SMF draft states that use of the SMF Type Message TLV is optional and is used when the router runs NHDP. But the draft does not clearly state if the use of the SMF Address Block TLV is also optional.

2. Is it useful to track the effectiveness of the coverage of the current RSSA? Is it possible to track this?

3. Complete the security analysis and section.

4. Cleanup all the [TODOs] from the MIB template.

Appendix C.

```
*************************************************************
* Note to the RFC Editor (to be removed prior to publication) *
*                                                             *
* 1) The reference to RFCXXXX within the DESCRIPTION clauses  *
* of the MIB module point to this draft and are to be         *
* assigned by the RFC Editor.                                 *
*                                                             *
* 2) The reference to RFCXXX2 throughout this document point  *
* to the current draft-ietf-manet-smf-xx.txt.  This           *
* need to be replaced with the XXX RFC number.                *
*                                                             *
*************************************************************
```

Authors' Addresses

Robert G. Cole
US Army CERDEC
328 Hopkins Road, Bldg 245
Aberdeen Proving Ground, Maryland 21005
USA

Phone: +1 410 278 6779
EMail: robert.g.cole@us.army.mil
URI: http://www.cs.jhu.edu/~rgcole/

Joseph Macker
Naval Research Laboratory
Washington, D.C. 20375
USA

EMail: macker@itd.nrl.navy.mil

Brian Adamson
Naval Research Laboratory
Washington, D.C. 20375
USA

EMail: adamson@itd.nrl.navy.mil

Sean Harnedy
Booz Allen Hamilton
333 City Boulevard West
Orange, CA 92868
USA

EMail: harnedy_sean@bah.com