Internet Draft
Document: <draft-zinman-trip-mib-00.txt>
Category: Informational

D. Walker
D. Zinman
SS8 Networks, Inc.

Management Information Base for Telephony Routing over IP (TRIP)

Status of this Memo

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 a set of managed objects that are used to manage for Telephony Routing over IP (TRIP) [17] devices.

Since TRIP [17] is modelled after the Border Gateway Protocol (BGP-4) [20], the managed objects for TRIP are also modelled after RFC1657 - Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP-4) using SMIv2 [21].

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 a set of managed objects that are used to schedule management operations periodically or at specified dates and times.

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 [16].

2. The SNMP Management Framework

The SNMP Management Framework presently consists of five major components:

- An overall architecture, described in RFC 2271 [1].

- Mechanisms for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIv1 and described in STD16, RFC 1155 [2], STD 16, RFC 1212 [3] and RFC 1215 [4]. The second version, called SMIv2, is described in STD 58, RFC 2578 [5], RFC 2579 [6] and RFC 2580 [7].

- Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in RFC 1157 [8]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901 [9] and RFC 1906 [10]. The third version of the message protocol is called SNMPv3 and described in RFC 2272 [11] and RFC 2274 [12].

- Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in STD 15, RFC 1157 [8]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [13].

- A set of fundamental applications described in RFC 2273 [14] and the view-based access control mechanism described in RFC 2275 [15].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the mechanisms defined in the SMI.

This memo specifies a MIB module that is compliant to the SMIv2. A MIB conforming to the SMIv1 can be produced through the appropriate translations. The resulting translated MIB must be semantically equivalent, except where objects or events are omitted because no translation is possible (use of Counter64). Some machine readable information in SMIv2 will be converted into textual descriptions in SMIv1 during the translation process. However, this loss of machine readable information is not considered to change the semantics of the MIB.

3. Overview

Telephony Routing over IP (TRIP) [17] is an inter-domain application-layer control protocol that exchanges information with other TRIP gateways to provide efficiant IP telephony routing. This MIB provides some managed objects for SIP devices defined in draft-ietf-iptel-trip-03.txt.

4. Structure of TRIP MIB

This MIB utilizes the framework described in rfc2788 [19] for management of multiple instances of TRIP from a single entity. The Network Services Monitoring MIB applTable will be populated with entries corresponding to each TRIP entity in the system. Each entity will then have an applIndex associated with it. The value assigned to applIndex will represent the distinct instance of TRIP.

The MIB defines some system-wide scalar objects local to the TRIP instance, as well as 5 tables: the Trip Peer Table, the Trip Peer Stats Table, the Trip Route Table, the Trip ITAD Topology Table, and the Trip ITAD Topology ID Table. The Trip Peer Table contains information about the state and current activity of the connections with TRIP peers. The Trip Peer Stats Table augments the Trip Peer Table and contains statistics related to the connections with TRIP peers. The Trip Route Table contains information on the route to a peers destination. The Trip ITAD Topology Table contains information on the sequence of link connections between peers within an ITAD. The Trip ITAD Topology ID Table is a subtable of the Trip ITAD Topology Table and contains the list of location servers within the ITAD domain that the instance of this trip ITAD Topology currently peering.

4.1 Textual Conventions

The data types TripItad and TripId are used as textual conventions in this document. A TRIP ITAD is described in [17]. A TRIP ID is used as a distinct identifier for a TRIP table entity. These textual conventions have NO effect on either the syntax nor the semantics of any managed object. Objects defined using these conventions are always encoded by means of the rules that define their primitive type. Hence, no changes to the SMI or the SNMP are necessary to accommodate these textual conventions which are adopted merely for the convenience of readers.

5. TRIP MIB

```
   TRIP-MIB DEFINITIONS ::= BEGIN

   IMPORTS
       MODULE-IDENTITY,
       OBJECT-TYPE,
       NOTIFICATION-TYPE,
       Unsigned32,
       Integer32,
       Gauge32,
       IpAddress,
       Counter32,
       mib-2
           FROM SNMPv2-SMI

       TEXTUAL-CONVENTION,
       DisplayString,
```

```
       DateAndTime,
       TruthValue,
       RowStatus
           FROM SNMPv2-TC

       OBJECT-GROUP,
       NOTIFICATION-GROUP
           FROM SNMPv2-CONF

       applIndex
            FROM NETWORK-SERVICES-MIB;

       tripMIB MODULE-IDENTITY
           LAST-UPDATED "200010200000Z"
           ORGANIZATION "IETF TRIP Working Group"
           CONTACT-INFO
           "Co-editor  Dave Walker
                       SS8 Networks, Inc.
            postal:    80 Hines Road
```

```
                       Ottawa, ON  K2K 2T8
                       Canada
            email:     drwalker@ss8networks.com
            phone:     +1 613 592 2100

            Co-editor  David Zinman
                       SS8 Networks, Inc.
            postal:    80 Hines Road
                       Ottawa, ON  K2K 2T8
                       Canada
            email:     david@ss8networks.com
            phone:     +1 613 592 2100"

       DESCRIPTION
           "The MIB module describing Telephony Routing
            Information Protocol (TRIP)"
       REVISION      "200008240000Z"
       DESCRIPTION
               "The initial revision of this MIB module was
                published as RFC xxx."
       ::= { mib-2 9996 } -- Temporary, until assigned

      --
      -- Textual Conventions
      --
      TripItad ::= TEXTUAL-CONVENTION
              STATUS current
              DESCRIPTION
                   "The values for identifying the IP Telephony
                   Administrative Domain."
              SYNTAX Integer32 (1..65534)

      TripId ::= TEXTUAL-CONVENTION
              STATUS current
```

```
              DESCRIPTION
                   "The range of legal values for a TRIP
                   Identifier."
              SYNTAX Integer32 (1..65534)

   trip                OBJECT IDENTIFIER ::= { tripMIB 1 }
   tripConformance     OBJECT IDENTIFIER ::= { tripMIB 2 }
   tripGroups          OBJECT IDENTIFIER ::= { tripConformance  1 }

       tripCfgTable OBJECT-TYPE
           SYNTAX     SEQUENCE OF TripCfgEntry
           MAX-ACCESS not-accessible
           STATUS     current
           DESCRIPTION
               "This table contains the common configuration objects
                applicable to all TRIP entities.  Each row represents
                those objects for a particular TRIP entity present in
                this system. The instances of TRIP entities are
                uniquely identified by applIndex."
           ::= { trip 1 }

       tripCfgEntry OBJECT-TYPE
           SYNTAX     TripCfgEntry
           MAX-ACCESS not-accessible
           STATUS     current
           DESCRIPTION
               "A row of common configuration."
           INDEX { applIndex }
           ::= { tripCfgTable 1 }

       TripCfgEntry ::=
           SEQUENCE {
               tripProtocolVersion               Integer32,
               tripLocalItad                     TripItad,
               tripIdentifier                    IpAddress,
               tripAdminStatus                   INTEGER,
               tripLocalAddr                     IpAddress,
               tripLocalPort                     Integer32,
               tripHoldTimeConfigured            Integer32,
               tripKeepAliveConfigured           Integer32,
               tripMinItadOriginationInterval    Integer32,
               tripMinRouteAdvertisementInterval Integer32,
               tripMaxPurgeTime                  Integer32,
               tripDisableTime                   Integer32
          }

       tripProtocolVersion    OBJECT-TYPE
           SYNTAX     Integer32 (1..255)
           MAX-ACCESS read-only
           STATUS     current
           DESCRIPTION
               "This object will reflect the version of TRIP
               supported by this system.  It follows the same
```

```
               format as TRIP version information contained
               in the TRIP messages generated by this TRIP entity
               as dictated by draft-ietf-iptel-trip-03.txt."
           ::= { tripCfgEntry 1 }
```

```
tripLocalItad   OBJECT-TYPE
    SYNTAX      TripItad
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The Local IP Telephony Administrative domain."
    ::= { tripCfgEntry 2 }

tripIdentifier   OBJECT-TYPE
    SYNTAX      IpAddress
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The object that identifies this TRIP Client. This
        object is the default for the tripPeerIdentifier value."
    ::= { tripCfgEntry 3 }

tripAdminStatus OBJECT-TYPE
    SYNTAX      INTEGER {
                    up(1),
                    down(2)
                }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The desired TRIP state."
    ::= { tripCfgEntry 4 }

tripLocalAddr OBJECT-TYPE
    SYNTAX      IpAddress
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The IP address of this entry's TRIP peer connection."
    ::= { tripCfgEntry 5 }

tripLocalPort OBJECT-TYPE
    SYNTAX      Integer32 (1..65535)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The local port that this entry's TRIP peer is using."
    ::= { tripCfgEntry 7 }

tripHoldTimeConfigured OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    UNITS       "Seconds"
    MAX-ACCESS  read-write
```

```
           STATUS      current
           DESCRIPTION
               "Specifies the maximum number of seconds that may
               elapse between the receipt of successive keepalive
               or update message by the sender."
           DEFVAL { 90 }
           ::= { tripCfgEntry 8 }

       tripKeepAliveConfigured OBJECT-TYPE
           SYNTAX      Integer32 (1..2147483647)
           UNITS       "Seconds"
           MAX-ACCESS  read-write
           STATUS      current
           DESCRIPTION
               "Specifies the amount of time that must elapse between
               keep alive messages."
           DEFVAL { 30 }
           ::= { tripCfgEntry 9 }

       tripMinItadOriginationInterval OBJECT-TYPE
           SYNTAX      Integer32 (1..2147483647)
           UNITS       "Seconds"
           MAX-ACCESS  read-write
           STATUS      current
           DESCRIPTION
               "Amount of time that must elapse between advertisement
               of update message that report changes within the
               Location Server's own ITAD."
           DEFVAL { 15 }
           ::= { tripCfgEntry 10 }

       tripMinRouteAdvertisementInterval OBJECT-TYPE
           SYNTAX      Integer32 (1..2147483647)
           UNITS       "Seconds"
           MAX-ACCESS  read-write
           STATUS      current
           DESCRIPTION
               "Specifies minimal interval between successive
               advertisement to a particular destination from an LS."
           DEFVAL { 30 }
           ::= { tripCfgEntry 11 }

       tripMaxPurgeTime OBJECT-TYPE
           SYNTAX      Integer32 (1..65535)
           UNITS       "Seconds"
           MAX-ACCESS  read-write
           STATUS      current
           DESCRIPTION
               "Indicate the interval that the location server must
               maintain routes marked as withdrawn in its database."
           ::= { tripCfgEntry 12 }

       tripDisableTime OBJECT-TYPE
```

```
           SYNTAX      Integer32 (1..65535)
           UNITS       "Seconds"
           MAX-ACCESS  read-write
           STATUS      current
           DESCRIPTION
               "Indicate the interval that the TRIP module of the
                location server must be disabled while routes
                originated by this location server with high
                sequence numbers can be removed."
           ::= { tripCfgEntry 13 }


   --
   -- TripPeerTable
   --
       tripPeerTable   OBJECT-TYPE
           SYNTAX      SEQUENCE OF TripPeerEntry
           MAX-ACCESS  not-accessible
           STATUS      current
           DESCRIPTION
               "The TRIP peer table. This table contains one entry per
               TRIP peer, and information about the connection with
               the peer."
           ::= { trip 2 }

       tripPeerEntry OBJECT-TYPE
           SYNTAX      TripPeerEntry
           MAX-ACCESS  not-accessible
           STATUS      current
           DESCRIPTION
               "Entry containing information about the connection with
               a TRIP peer."
           INDEX   { applIndex, tripPeerRemoteAddr }
             ::= {tripPeerTable 1}

       TripPeerEntry ::= SEQUENCE {
           tripPeerRemoteAddr                    IpAddress,
           tripPeerIdentifier                    TripId,
           tripPeerState                         INTEGER,
           tripPeerAdminStatus                   INTEGER,
           tripPeerNegotiatedVersion             Integer32,
           tripPeerOpMode                        INTEGER,
           tripPeerSupportedProtocol             INTEGER,
           tripPeerAddressFamily                 INTEGER,
           tripPeerRemotePort                    Integer32,
           tripPeerRemoteItad                    TripItad,
           tripPeerConnectRetryInterval          Integer32,
           tripPeerMaxRetryInterval              Integer32,
           tripPeerHoldTime                      Integer32,
           tripPeerKeepAlive                     Integer32,
           tripPeerHoldTimeConfigured            Integer32,
           tripPeerKeepAliveConfigured           Integer32,
           tripPeerMinItadOriginationInterval    Integer32,
```

```
           tripPeerMinRouteAdvertisementInterval Integer32,
           tripPeerMaxPurgeTime                  Integer32,
           tripPeerDisableTime                   Integer32,
           tripPeerRowStatus                     RowStatus
       }

       tripPeerRemoteAddr OBJECT-TYPE
           SYNTAX      IpAddress
           MAX-ACCESS  not-accessible
           STATUS      current
           DESCRIPTION
               "The remote IP address of this entry's TRIP peer."
           ::= { tripPeerEntry 1 }

       tripPeerIdentifier OBJECT-TYPE
           SYNTAX      TripId
           MAX-ACCESS  read-only
           STATUS      current
           DESCRIPTION
               "TRIP identifier of this entry's TRIP peer. The default
               is the value of the identifier of the remote."
           ::= { tripPeerEntry 2 }

       tripPeerState OBJECT-TYPE
           SYNTAX      INTEGER {
                           idle(1),
                           connect(2),
                           active(3),
                           openSent(4),
                           openConfirm(5),
                           established(6)
```

```
                       }
           MAX-ACCESS  read-only
           STATUS      current
           DESCRIPTION
               "TRIP Peer Finite State Machine state."
           ::= { tripPeerEntry 3 }

       tripPeerAdminStatus OBJECT-TYPE
           SYNTAX      INTEGER {
                           up(1),
                           down(2)
                       }
           MAX-ACCESS  read-create
           STATUS      current
           DESCRIPTION
               "The desired TRIP connection state."
           ::= { tripPeerEntry 4 }

       tripPeerNegotiatedVersion OBJECT-TYPE
           SYNTAX      Integer32 (1..255)
           MAX-ACCESS  read-only
           STATUS      current
```

```
           DESCRIPTION
               "The negotiated version of TRIP running between this
               local entity and this peer."
           ::= { tripPeerEntry 5 }

       tripPeerOpMode OBJECT-TYPE
           SYNTAX      INTEGER {
                   sendOnly(1),
                   receiveOnly(2),
                   sendReceive(3)
                   }
           MAX-ACCESS  read-only
           STATUS      current
           DESCRIPTION
               "The operational mode of this peer."
           ::= { tripPeerEntry 6 }

       tripPeerSupportedProtocol OBJECT-TYPE
           SYNTAX      INTEGER {
                   other(1),
                   sip(2),
                   h323(3)
                   }
           MAX-ACCESS  read-only
           STATUS      current
           DESCRIPTION
               "The protocol that this peer is using."
           ::= { tripPeerEntry 7 }

       tripPeerAddressFamily OBJECT-TYPE
           SYNTAX      INTEGER {
                   other(1),
                   pots(2),
                   routed(3)
                   }
           MAX-ACCESS  read-only
           STATUS      current
           DESCRIPTION
               "The address family that this peer belongs."
           ::= { tripPeerEntry 8 }

       tripPeerRemotePort OBJECT-TYPE
           SYNTAX      Integer32 (1..65535)
           MAX-ACCESS  read-only
           STATUS      current
           DESCRIPTION
               "The remote ports for the TCP connection between the
               TRIP peers."
           ::= { tripPeerEntry 9 }

       tripPeerRemoteItad OBJECT-TYPE
           SYNTAX      TripItad
           MAX-ACCESS  read-only
```

```
           STATUS      current
           DESCRIPTION
               "The IP Telephony Administrative domain of this peer."
           ::= { tripPeerEntry 10 }

       tripPeerConnectRetryInterval OBJECT-TYPE
           SYNTAX      Integer32 (0..2147483647)
           UNITS       "Seconds"
           MAX-ACCESS  read-create
           STATUS      current
           DESCRIPTION
               "Specifies the initial amount of time that will elapse
               between connection retry. This value should double
               after each attempt up to the value of
               tripPeerMaxRetryInterval."
           DEFVAL      { 60 }
           ::= { tripPeerEntry 11 }

       tripPeerMaxRetryInterval OBJECT-TYPE
           SYNTAX      Integer32 (0..2147483647)
           UNITS       "Seconds"
           MAX-ACCESS  read-create
           STATUS      current
           DESCRIPTION
               "Specifies the maximum amount of time that will elapse
               between connection retries."
           DEFVAL      { 360 }
           ::= { tripPeerEntry 12 }


       tripPeerHoldTime OBJECT-TYPE
           SYNTAX      Integer32 (1..2147483647)
           UNITS       "Seconds"
           MAX-ACCESS  read-only
           STATUS      current
           DESCRIPTION
               "The time interval in seconds for the hold timer that
               is established with the peer. The value of this object
               is the smaller of the values in
               tripPeerHoldTimeConfigured and the hold time received
               in the open message."
           ::= { tripPeerEntry 13 }

       tripPeerKeepAlive OBJECT-TYPE
           SYNTAX      Integer32 (1..2147483647)
           UNITS       "Seconds"
           MAX-ACCESS  read-only
           STATUS      current
           DESCRIPTION
               "Specifies the amount of time that must elapse between
               keep alive messages."
           ::= { tripPeerEntry 14 }
```

```
       tripPeerHoldTimeConfigured OBJECT-TYPE
           SYNTAX      Integer32 (0..65535)
           UNITS       "Seconds"
           MAX-ACCESS  read-create
           STATUS      current
           DESCRIPTION
               "Specifies the maximum time that may elapse between the
```

```
                receipt of successive keepalive or update message."
            DEFVAL { 90 }
            ::= { tripPeerEntry 15 }

        tripPeerKeepAliveConfigured OBJECT-TYPE
            SYNTAX      Integer32 (1..2147483647)
            UNITS       "Seconds"
            MAX-ACCESS  read-create
            STATUS      current
            DESCRIPTION
                "Specifies the amount of time that must elapse between
                keep alive messages."
            DEFVAL { 30 }
            ::= { tripPeerEntry 16 }

        tripPeerMinItadOriginationInterval OBJECT-TYPE
            SYNTAX      Integer32 (0..2147483647)
            UNITS       "Seconds"
            MAX-ACCESS  read-create
            STATUS      current
            DESCRIPTION
                "Amount of time that must elapse between advertisement
                of update message that report changes within the Location
                Server's own ITAD."
            DEFVAL { 15 }
            ::= { tripPeerEntry 17 }

        tripPeerMinRouteAdvertisementInterval OBJECT-TYPE
            SYNTAX      Integer32 (1..2147483647)
            UNITS       "Seconds"
            MAX-ACCESS  read-create
            STATUS      current
            DESCRIPTION
                "Specifies minimal interval between successive
                advertisement to a particular destination from an LS."
            DEFVAL { 30 }
            ::= { tripPeerEntry 18 }

        tripPeerMaxPurgeTime OBJECT-TYPE
            SYNTAX      Integer32 (1..65535)
            UNITS       "Seconds"
            MAX-ACCESS  read-create
            STATUS      current
            DESCRIPTION
                "Indicate the interval that the location server must
                maintain routes marked as withdrawn in its database."
            ::= { tripPeerEntry 19 }

        tripPeerDisableTime OBJECT-TYPE
            SYNTAX      Integer32 (1..65535)
            UNITS       "Seconds"
            MAX-ACCESS  read-create
            STATUS      current
            DESCRIPTION
                "Indicate the interval that the TRIP module of the
                location server must be disabled while routes
                originated by this location server with high sequence
                numbers can be removed."
            ::= { tripPeerEntry 20 }

        tripPeerRowStatus OBJECT-TYPE
            SYNTAX      RowStatus
            MAX-ACCESS  read-create
            STATUS      current
            DESCRIPTION
                "This object is used to create and delete rows in the
                tripPeerTable."
            ::= { tripPeerEntry 21 }


    --
    -- TripPeerStatsTable
    --
        tripPeerStatsTable   OBJECT-TYPE
            SYNTAX      SEQUENCE OF TripPeerStatsEntry
            MAX-ACCESS  not-accessible
            STATUS      current
            DESCRIPTION
                "The TRIP peer table. This table contains one entry per
                TRIP peer, and information about the connection with
                the peer."
            ::= { trip 3 }

        tripPeerStatsEntry OBJECT-TYPE
            SYNTAX      TripPeerStatsEntry
            MAX-ACCESS  not-accessible
            STATUS      current
            DESCRIPTION
                "Entry containing information about the connection with
                a TRIP peer."
            AUGMENTS { tripPeerEntry }
              ::= { tripPeerStatsTable 1 }

        TripPeerStatsEntry ::= SEQUENCE {
            tripPeerInUpdates                   Counter32,
            tripPeerOutUpdates                  Counter32,
            tripPeerInTotalMessages             Counter32,
            tripPeerOutTotalMessages            Counter32,
            tripPeerLastError                   OCTET STRING,
```

```
           tripPeerFsmEstablishedTransitions   Counter32,
           tripPeerFsmEstablishedTime          DateAndTime,
           tripPeerInUpdateElapsedTime         Gauge32
       }

        tripPeerInUpdates OBJECT-TYPE
           SYNTAX      Counter32
           MAX-ACCESS  read-only
           STATUS      current
           DESCRIPTION
               "The number of TRIP update messages received from this
               peer since the last restart of this location server."
           ::= { tripPeerStatsEntry 1 }

       tripPeerOutUpdates OBJECT-TYPE
           SYNTAX      Counter32
           MAX-ACCESS  read-only
           STATUS      current
           DESCRIPTION
               "The number of TRIP update messages transmitted to
               this peer since the last restart of this location
               server."
           ::= { tripPeerStatsEntry 2 }

       tripPeerInTotalMessages OBJECT-TYPE
           SYNTAX      Counter32
           MAX-ACCESS  read-only
           STATUS      current
           DESCRIPTION
               "The total number of TRIP messages received to the
               remote peer on this connection since the last restart
               of this location server."
           ::= { tripPeerStatsEntry 3 }

       tripPeerOutTotalMessages OBJECT-TYPE
           SYNTAX      Counter32
           MAX-ACCESS  read-only
           STATUS      current
           DESCRIPTION
               "The total number of outgoing TRIP messages sent since
               the last restart of this location server."
           ::= { tripPeerStatsEntry 4 }
       tripPeerLastError OBJECT-TYPE
           SYNTAX      OCTET STRING (SIZE(2))
           MAX-ACCESS  read-only
           STATUS      current
           DESCRIPTION
               "Notification message of TRIP error.
               The first octet signifies the error
               code. The second octet signifies the
               error subcode.
```

```
               1 - message-header
               2 - open-message
               3 - update-message
               4 - hold-timer-expired
               5 - finite-state-machine
               6 - cease

               The sub error code associated with
               error code. The meaning of this
               value is dependent on the value of
               the first octet.

               Message Header (1) Error Subcodes:
               1 - Bad Message Length.
               2 - Bad Message Type.

               OPEN Message (2) Error Subcodes:
               1 - Unsupported Version Number.
               2 - Bad Peer ITAD.
               3 - Bad TRIP Identifier.
               4 - Unsupported Optional Parameter.
               5 - Unacceptable Hold Time.
               6 - Unsupported Capability.

               UPDATE Message (3) Error Subcodes:
               1 - Malformed Attribute List.
               2 - Unrecognized Well-known Attribute.
               3 - Missing Well-known Mandatory Attribute.
               4 - Attribute Flags Error.
               5 - Attribute Length Error.
               6 - Invalid Attribute."
          ::= { tripPeerStatsEntry 5 }

       tripPeerFsmEstablishedTransitions OBJECT-TYPE
           SYNTAX      Counter32
           MAX-ACCESS  read-only
           STATUS      current
           DESCRIPTION
               "The number of times the TRIP peer has transitioned into
               the established state since the last restart of this
               location server."
           ::= { tripPeerStatsEntry 6 }

       tripPeerFsmEstablishedTime OBJECT-TYPE
           SYNTAX      DateAndTime
           MAX-ACCESS  read-only
           STATUS      current
           DESCRIPTION
               "Indicates how long in seconds this peer has been in the
               established state."
           ::= { tripPeerStatsEntry 7 }

       tripPeerInUpdateElapsedTime OBJECT-TYPE
```

```
           SYNTAX      Gauge32
           MAX-ACCESS  read-only
           STATUS      current
           DESCRIPTION
               "Elapsed time in seconds since the last TRIP update
               message was received from the peer."
           ::= { tripPeerStatsEntry 8 }


   -- TRIP Received Route Table.  This table contains
   -- all routes from all sources. Each entry consists
   -- of a route and its associated path attributes.

       tripRouteTable OBJECT-TYPE
           SYNTAX      SEQUENCE OF TripRouteEntry
           MAX-ACCESS  not-accessible
           STATUS      current
           DESCRIPTION
               "The TRIP route table containing information
               about routes to the called destinations received from
               all TRIP peers."
           ::= { trip 4 }

       tripRouteEntry OBJECT-TYPE
           SYNTAX      TripRouteEntry
           MAX-ACCESS  not-accessible
           STATUS      current
           DESCRIPTION
```

```
               "Information about a route to a called destination."
           INDEX { applIndex,
                   tripRouteAppProtocol,
                   tripRouteAddress,
                   tripRouteAddressLen,
                   tripRoutePeer }
           ::= { tripRouteTable 1 }

       TripRouteEntry ::= SEQUENCE {
                   tripRouteAppProtocol                 INTEGER,
                   tripRouteAddress                     OCTET STRING,
                   tripRouteAddressLen                  Integer32,
                   tripRoutePeer                        IpAddress,
                   tripRouteAddressFamily               INTEGER,
                   tripRouteCommunity                   OCTET STRING,
                   tripRouteAddressSequenceNumber       Integer32,
                   tripRouteAddressOriginatorId         TripItad,
                   tripRouteNextHopServer               DisplayString,
                   tripRouteNextHopServerPort           Integer32,
                   tripRouteNextHopServerItad           TripItad,
                   tripRouteMultiExitDisc               Unsigned32,
                   tripRouteLocalPref                   Unsigned32,
                   tripRouteAdvertisementPathSegment    OCTET STRING,
                   tripRoutePathSegment                 OCTET STRING,
                   tripRouteAtomicAggregate             TruthValue,
```

```
                   tripRouteBest                        TruthValue,
                   tripRouteUnknown                     OCTET STRING
               }

       tripRouteAppProtocol OBJECT-TYPE
           SYNTAX INTEGER {
                       sip(1),
                       h323q931(2),
                       h323ras(3),
                       h323annexg(4),
                       other(100)
                  }
           MAX-ACCESS  not-accessible
           STATUS      current
           DESCRIPTION
               "The protocol for which this routing table is
               maintained."
           ::= { tripRouteEntry 1 }

       tripRouteAddress OBJECT-TYPE
```

```
           SYNTAX      OCTET STRING (SIZE(0..31))
           MAX-ACCESS  not-accessible
           STATUS      current
           DESCRIPTION
               "Destination address in E164 format."
           ::= { tripRouteEntry 2 }

       tripRouteAddressLen OBJECT-TYPE
           SYNTAX      Integer32 (1..255)
           MAX-ACCESS  not-accessible
           STATUS      current
           DESCRIPTION
               "Length of the destination address."
           ::= { tripRouteEntry 3 }

       tripRoutePeer OBJECT-TYPE
           SYNTAX      TripId
           MAX-ACCESS  not-accessible
           STATUS      current
           DESCRIPTION
               "The identifier of the peer where the route information
               was learned."
           ::= { tripRouteEntry 4 }

       tripRouteAddressFamily OBJECT-TYPE
           SYNTAX      INTEGER {
                          pots(1),
                          routedNumber(2)
                       }
           MAX-ACCESS  read-only
           STATUS      current
           DESCRIPTION
               "Specifies the type of address for the destination
```

```
               route."
           ::= { tripRouteEntry 5 }

       tripRouteCommunity OBJECT-TYPE
           SYNTAX      OCTET STRING (SIZE(4))
           MAX-ACCESS  read-only
           STATUS      current
           DESCRIPTION
               "The community that has been bound to this route."
           ::= { tripRouteEntry 6 }

       tripRouteAddressSequenceNumber OBJECT-TYPE
```

```
            SYNTAX      Integer32 (1..2147483647)
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                "Indicates the version of the destination route
                originated by the location server identified by
                tripRouteAddressOriginatorId intra-domain
                attribute."
            ::= { tripRouteEntry 7 }

        tripRouteAddressOriginatorId OBJECT-TYPE
            SYNTAX      TripItad
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                "This is an intra-domain attribute indicating the
                internal location server that originated the route
                into the ITAD."
            ::= { tripRouteEntry 8 }

        tripRouteNextHopServer OBJECT-TYPE
            SYNTAX      DisplayString
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                "Indicates the next hop that messages of a given
                protocol destined for tripRouteAddress should
                be sent to."
            ::= { tripRouteEntry 9 }

        tripRouteNextHopServerPort OBJECT-TYPE
            SYNTAX      Integer32 (1..2147483647)
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                "The port of the next hop server that this route
                will use."
            ::= { tripRouteEntry 10 }

        tripRouteNextHopServerItad OBJECT-TYPE
            SYNTAX      TripItad
```

```
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                "Indicates the domain of the next hop."
```

```
        ::= { tripRouteEntry 11 }

    tripRouteMultiExitDisc OBJECT-TYPE
        SYNTAX      Unsigned32 (1..2147483647)
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "Used to descriminate between multiple exit points to
            an adjacent ITAD."
        ::= { tripRouteEntry 12 }

    tripRouteLocalPref OBJECT-TYPE
        SYNTAX      Unsigned32 (1..2147483647)
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "Indicated the originating TRIP's degree of preference
            for an advertised route destination."
        ::= { tripRouteEntry 13 }

    tripRouteAdvertisementPathSegment OBJECT-TYPE
        SYNTAX      OCTET STRING (SIZE(2..255))
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The sequence of advertisement path segments an update
            message has passed.

            This object is probably best represented as SEQUENCE OF
            INTEGER.  For SMI compatibility, though, it is
            represented as OCTET STRING. Each ITAD is represented as
            a pair of octets according to the following algorithm:

            first-byte-of-pair = ItadNumber / 256;
            second-byte-of-pair = ItadNumber & 255;"
        ::= { tripRouteEntry 14 }

    tripRoutePathSegment OBJECT-TYPE
        SYNTAX      OCTET STRING (SIZE(2..255))
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "A sequence of ITAD segment indicating the actual path
            to the destination.

            This object is probably best represented as SEQUENCE OF
            INTEGER.  For SMI compatibility, though, it is
            represented as OCTET STRING.  Each ITAD is represented as
            a pair of octets according to the following algorithm:
            first-byte-of-pair = ItadNumber / 256;
            second-byte-of-pair = ItadNumber & 255;"
        ::= { tripRouteEntry 15 }

    tripRouteAtomicAggregate OBJECT-TYPE
        SYNTAX      TruthValue
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "Indicates whether or not a system has selected a less
            specific route without selecting a more specfic route."
        ::= { tripRouteEntry 16 }

    tripRouteBest OBJECT-TYPE
        SYNTAX      TruthValue
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "An indication of whether this route was chosen as the
            best TRIP route."
        ::= { tripRouteEntry 17 }

    tripRouteUnknown OBJECT-TYPE
        SYNTAX      OCTET STRING (SIZE(0..255))
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "One or more attributes not understood by this location
            server."
        ::= { tripRouteEntry 18 }

--
-- tripItadTopologyTable
--

    tripItadTopologyTable OBJECT-TYPE
        SYNTAX      SEQUENCE OF TripItadTopologyEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The sequence of link connections between peers within
            an ITAD."
        ::= { trip 5 }

    tripItadTopologyEntry OBJECT-TYPE
        SYNTAX      TripItadTopologyEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "Information about a peer of the location server
            identified by tripOriginatorIdentifier."
        INDEX { applIndex, tripOriginatorIdentifier }
```

```
           ::= { tripItadTopologyTable 1 }

       TripItadTopologyEntry ::= SEQUENCE {
                   tripOriginatorIdentifier    TripItad,
                   tripSequenceNumber          Integer32
               }

       tripOriginatorIdentifier OBJECT-TYPE
           SYNTAX      TripItad
           MAX-ACCESS  not-accessible
           STATUS      current
           DESCRIPTION
               "Indicates the internal location server that originated
               the ITAD topology information into the ITAD."
           ::= { tripItadTopologyEntry 1 }

       tripSequenceNumber OBJECT-TYPE
           SYNTAX      Integer32 (1..2147483647)
           MAX-ACCESS  read-only
           STATUS      current
           DESCRIPTION
               "Indicates the version of the ITAD topology
               originated by the location server identified by
               tripOriginatorIdentifier."
           ::= { tripItadTopologyEntry 2 }


   --
   -- tripItadTopologyIpTable
   --

       tripItadTopologyIdTable OBJECT-TYPE
           SYNTAX      SEQUENCE OF TripItadTopologyIdEntry
           MAX-ACCESS  not-accessible
           STATUS      current
           DESCRIPTION
               "The list of other location servers within the ITAD
               domain that the location server identified by
               tripOriginatorIdentifier is currently peering."
           ::= { trip 6 }
       tripItadTopologyIdEntry OBJECT-TYPE
           SYNTAX      TripItadTopologyIdEntry
           MAX-ACCESS  not-accessible
           STATUS      current
           DESCRIPTION
               "Information about a peer to the location server
               identified by tripOriginatorIdentifier."
           INDEX { applIndex,
                   tripOriginatorIdentifier,
                   tripItadTopologyIdIndex }
           ::= { tripItadTopologyIdTable 1 }
```

```
       TripItadTopologyIdEntry ::= SEQUENCE {
                   tripItadTopologyIdIndex       TripId,
                   tripItadTopologyIdIdentifier  TripId
               }

       tripItadTopologyIdIndex OBJECT-TYPE
           SYNTAX      TripId
           MAX-ACCESS  not-accessible
           STATUS      current
           DESCRIPTION
               "The index into this entry. This will be the same value
               as tripItadTopologyIdentifier."
           ::= { tripItadTopologyIdEntry 1 }

       tripItadTopologyIdIdentifier OBJECT-TYPE
           SYNTAX      TripId
           MAX-ACCESS  read-only
           STATUS      current
           DESCRIPTION
               "Indicates the other location servers within the ITAD
               domain that this location server identified by
               tripOriginatorIdentifier is currently peering."
           ::= { tripItadTopologyIdEntry 2 }


   -- *************
   -- Notifications
   -- *************

       tripTraps   OBJECT IDENTIFIER ::= { trip 0 }

       tripEstablished NOTIFICATION-TYPE
           OBJECTS { tripPeerLastError,
                     tripPeerState
                   }
           STATUS  current
           DESCRIPTION
               "The TRIP Established event is generated when the TRIP
               FSM enters the ESTABLISHED state."
           ::= { tripTraps 1 }

       tripBackwardTransition NOTIFICATION-TYPE
           OBJECTS { tripPeerLastError,
                     tripPeerState
                   }
           STATUS  current
           DESCRIPTION
               "The TRIPBackwardTransition Event is generated when the
               TRIP FSM moves from a higher numbered state to a lower
               numbered state."
           ::= { tripTraps 2 }
```

```
   -- **********************
   -- Object and event groups
   -- **********************

       tripConfigGroup OBJECT-GROUP
           OBJECTS {
               tripProtocolVersion,
               tripLocalItad,
               tripIdentifier,
               tripAdminStatus,
               tripLocalAddr,
               tripLocalPort,
               tripHoldTimeConfigured,
               tripKeepAliveConfigured,
               tripMinItadOriginationInterval,
               tripMinRouteAdvertisementInterval,
               tripMaxPurgeTime,
               tripDisableTime,
               tripPeerRowStatus
           }
           STATUS current
           DESCRIPTION
               "The global objects for configuring trip."
           ::= { tripGroups 1 }


       tripPeerTableConfigGroup OBJECT-GROUP
```

```
        OBJECTS {
            tripPeerIdentifier,
            tripPeerState,
            tripPeerAdminStatus,
            tripPeerNegotiatedVersion,
            tripPeerOpMode,
            tripPeerSupportedProtocol,
            tripPeerAddressFamily,
            tripPeerRemotePort,
            tripPeerRemoteItad,
            tripPeerConnectRetryInterval,
            tripPeerMaxRetryInterval,
            tripPeerHoldTime,
            tripPeerKeepAlive,
            tripPeerHoldTimeConfigured,
            tripPeerKeepAliveConfigured,
            tripPeerMinItadOriginationInterval,
            tripPeerMinRouteAdvertisementInterval,
            tripPeerMaxPurgeTime,
            tripPeerDisableTime
            }

        STATUS current
        DESCRIPTION
            "The global objects for configuring the trip peer table."
        ::= { tripGroups 2 }
```

```
    tripPeerTableStatsGroup OBJECT-GROUP
        OBJECTS {
            tripPeerInUpdates,
            tripPeerOutUpdates,
            tripPeerInTotalMessages,
            tripPeerOutTotalMessages,
            tripPeerLastError,
            tripPeerFsmEstablishedTransitions,
            tripPeerFsmEstablishedTime,
            tripPeerInUpdateElapsedTime
            }

        STATUS current
        DESCRIPTION
            "The global statistics the trip peer table."
        ::= { tripGroups 3 }

    tripRouteGroup OBJECT-GROUP
```

```
           OBJECTS {
               tripRouteAddressFamily,
               tripRouteCommunity,
               tripRouteAddressSequenceNumber,
               tripRouteAddressOriginatorId,
               tripRouteNextHopServer,
               tripRouteNextHopServerPort,
               tripRouteNextHopServerItad,
               tripRouteMultiExitDisc,
               tripRouteLocalPref,
               tripRouteAdvertisementPathSegment,
               tripRoutePathSegment,
               tripRouteAtomicAggregate,
               tripRouteBest,
               tripRouteUnknown
               }

           STATUS current
           DESCRIPTION
               "The global objects for configuring route attribute."
           ::= { tripGroups 4 }

       tripItadTopologyGroup OBJECT-GROUP
           OBJECTS {
               tripSequenceNumber,
               tripItadTopologyIdIdentifier
               }
           STATUS current
           DESCRIPTION
               "The objects that define the ITAD topology."
           ::= { tripGroups 5 }


       tripNotificationGroup NOTIFICATION-GROUP
```

```
           NOTIFICATIONS {
               tripEstablished,
               tripBackwardTransition
           }
           STATUS  current
           DESCRIPTION
                "A collection of notifications defined for TRIP."
           ::= { tripGroups 6 }

   END
```

6. Security Considerations

There are a number of management objects defined in this MIB that have 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.

The managed objects in this MIB contain sensitive information since, collectively, they allow tracing and influencing of connections in TRIP devices and provide information of their connection characteristics.

It is thus important to control even GET access to these objects and possibly to even encrypt the values of these object when sending them over the network via SNMP. Not all versions of SNMP provide features for such a secure environment.

SNMPv1 by itself is not a secure environment. 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.

It is recommended that the implementers consider the security features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model RFC 2574 [18] and the View-based Access Control Model RFC 2575 [19] is recommended.

It is then a customer/user responsibility to ensure that the SNMP entity giving access to an instance of this MIB, 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.

7. Intellectual Property

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it

has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

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

8. Full Copyright Statement

9. References

[1] Harrington, D., Presuhn, R. and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", RFC 2271, January 1998.

[2] Rose, M. and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", STD 16, RFC 1155, May 1990.

[3] Rose, M. and K. McCloghrie, "Concise MIB Definitions", STD 16, RFC 1212, March 1991.

[4] Rose, M., "A Convention for Defining Traps for use with the SNMP", RFC 1215, March 1991.

[5] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[6] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[7] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

[8] Case, J., Fedor, M., Schoffstall, M. and J. Davin, "Simple Network Management Protocol", STD 15, RFC 1157, May 1990.

[9] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Introduction to Community-based SNMPv2", RFC 1901, January 1996.

[10] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1906, January 1996.

[11] Case, J., Harrington D., Presuhn R. and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", RFC 2272, January 1998.

[12] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", RFC 2274, January 1998.

[13] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", January 1996.

[14] Levi, D., Meyer, P. and B. Stewart, "SNMPv3 Applications", RFC 2273, January 1998

[15] Wijnen, B., Presuhn, R. and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", RFC 2275, January 1998.

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

[17] J. Rosenberg, H. Salama, and M. Squire, "Telephony Routing over IP (TRIP)", draft-ietf-iptel-trip-03.txt, work in progress.

[18] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", RFC 2574, April 1999.

[19] N. Freed, and S, Kille, "Network Services Monitoring MIB", RFC 2788, March 2000.

[20] Y. Rekhter and T. Li, "Border Gateway Protocol 4 (BGP-4),"
IETF RFC 1771, March 1995.

[21] S. Willis, J. Burruss, and J. Chu. "Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP-4) using SMIv2" IETF RFC 1657, July 1994.

10. Authors' Addresses

Dave Walker
SS8 Networks, Inc.
80 Hines Road
Kanata, ON K2K 2T8 Canada
email: drwalker@ss8networks.com
phone: +1 613 592 2100

David Zinman
SS8 Networks, Inc.
80 Hines Road
Kanata, ON K2K 2T8 Canada
email: david@ss8networks.com
phone: +1 613 592 2100