Internet Engineering Task Force
Internet Draft
Document: draft-zinman-trip-mib-01.txt
Expires: September 2001

D. Walker
D. Zinman
SS8 Networks, Inc
March 2001

Management Information Base for Telephony Routing over IP (TRIP) <draft-zinman-trip-mib-01.txt>

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) [2] devices.

Since TRIP [2] is modelled after the Border Gateway Protocol (BGP-4) [3], 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 [4].

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.

2. Conventions used in this document

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 [5].

3. The SNMP Management Framework

The SNMP Management Framework presently consists of five major components:

- An overall architecture, described in RFC 2271 [6].

- 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 STD 16, RFC 1155 [7], STD 16, RFC 1212 [8] and RFC 1215 [9]. The second version, called SMIv2, is described in STD 58, RFC 2578 [10], RFC 2579 [11] and RFC 2580 [12].

- Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in RFC 1157 [13]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901 [14] and RFC 1906 [15]. The third version of the message protocol is called SNMPv3 and described in RFC 2272 [16] and RFC 2274 [17].

- Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in STD 15, RFC 1157 [13]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [18].

- A set of fundamental applications described in RFC 2273 [19] and the view-based access control mechanism described in RFC 2275 [20].

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.

4. Overview

Telephony Routing over IP (TRIP) [2] is an inter-domain application-layer control protocol that exchanges information with other TRIP gateways to provide efficient IP telephony routing. This MIB provides some managed objects for TRIP devices defined in Telephony Routing over IP <draft-ietf-iptel-trip-04.txt> and Authentication Attribute for TRIP <draft-ietf-iptel-authen-00.txt> [21].

5. Structure of TRIP MIB

This MIB utilizes the framework described in RFC 2788 [22] 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 Authentication Mechanism Table, the Trip Peer Table, the Trip Peer Stats Table, the Trip Route Table, the Trip Route Community Table, the Trip ITAD Topology Table, and the Trip ITAD Topology ID Table.

The Trip Authentication Mechanism Table contains the authentication mechanism and the public key to broadcast.

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 Route Community Table contains information on the communities associated with each route.

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.

5.1. Textual Conventions

The data types TripItad and TripId are used as textual conventions in this document. A TRIP ITAD is described in [2]. A TRIP ID is used as a distinct identifier for a TRIP entity. A TripPublicKey defines a key broadcast in an authentication transaction. 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.

6. TRIP MIB

```
TRIP-MIB DEFINITIONS ::= BEGIN

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

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

```
        OBJECT-GROUP,
        NOTIFICATION-GROUP
            FROM SNMPv2-CONF

        InetAddressType,
        InetAddress
            FROM INET-ADDRESS-MIB

        applIndex
            FROM NETWORK-SERVICES-MIB;


    tripMIB MODULE-IDENTITY
       LAST-UPDATED "200102260000Z"
       ORGANIZATION "IETF IPTel Working Group"
       CONTACT-INFO
       "Co-editor  Dave Walker
                   SS8 Networks, Inc.
        postal:    495 March Road, Suite #500
                   Ottawa, ON, K2K 3G1 Canada
        email:     drwalker@ss8.com
        phone:     +1 613 592 2100

        Co-editor  David Zinman
                   SS8 Networks, Inc.
        postal:    495 March Road, Suite #500
                   Ottawa, ON, K2K 3G1 Canada
```



```
        email:     david@ss8.com
        phone:     +1 613 592 2100"

    DESCRIPTION
       "The MIB module describing Telephony Routing
        over IP (TRIP)"
    REVISION      "200102260000Z"
    DESCRIPTION
        "The initial revision of this MIB module was
        published as draft-zinman-trip-mib-00.txt."
    ::= { mib-2 } -- to be assigned by IANA

    --
    -- Textual Conventions
    --
    TripItad ::= TEXTUAL-CONVENTION
        STATUS current
        DESCRIPTION
```

```
        "The values for identifying the IP Telephony
        Administrative Domain."
     SYNTAX Unsigned32 (0..4294967295)

   TripId ::= TEXTUAL-CONVENTION
     STATUS current
     DESCRIPTION
        "The range of legal values for a TRIP Identifier."
     SYNTAX Unsigned32 (0..4294967295)

   TripPublicKey ::= TEXTUAL-CONVENTION
     STATUS current
     DESCRIPTION
          "A public key that is exchanged in an authentication
          transaction."
     SYNTAX OCTET STRING (SIZE(0..256))

   TripAppProtocol ::= TEXTUAL-CONVENTION
      STATUS current
      DESCRIPTION
          "The application protocol used for communication with
          TRIP Location Servers."
      SYNTAX INTEGER {
                sip(1),
                h323q931(2),
                h323ras(3),
                h323annexg(4),
                other(100)
              }

   TripAddressFamily ::= TEXTUAL-CONVENTION
      STATUS current
      DESCRIPTION
          "A type of address for a TRIP route."
      SYNTAX INTEGER {
```



```
                decimal(1),
                hexaDecimal(2),
                other(100)
              }

     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 LS present in
             this system. The instances of TRIP LS's 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                   TripId,
            tripAdminStatus                  INTEGER,
            tripLocalAddrIAddrType           InetAddressType,
            tripLocalAddr                    InetAddress,
            tripLocalPort                    Integer32,
            tripMinItadOriginationInterval   Integer32,
            tripMinRouteAdvertisementInterval Integer32,
            tripMaxPurgeTime                 Integer32,
            tripDisableTime                  Integer32,
            tripSendReceiveMode              INTEGER
       }

    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-04.txt."
```

```
    ::= { tripCfgEntry 1 }

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

tripIdentifier   OBJECT-TYPE
    SYNTAX      TripId
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The object that identifies this TRIP Client."
    ::= { tripCfgEntry 3 }

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

         up(1)  : The application is operating normally, and
                  is processing (receiving and possibly
                  issuing) TRIP requests and responses.

         down(2): The application is currently unable to
                  process TRIP messages."
    ::= { tripCfgEntry 4 }

tripLocalAddrIAddrType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The type of Inet Address of the tripLocalAddr."
    REFERENCE
        "RFC 2851, section 3."
    ::= { tripCfgEntry 5 }

tripLocalAddr OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-only
    STATUS      current
```

```
        DESCRIPTION
            "The IP address of this entry's TRIP peer connection."
        REFERENCE
            "RFC 2851, section 3."
        ::= { tripCfgEntry 6 }

    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 }

    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 { 30 }
        ::= { tripCfgEntry 8 }

    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 9 }

    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."
        DEFVAL { 10 }
```

```
          ::= { tripCfgEntry 10 }

      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."
          DEFVAL { 180 }
          ::= { tripCfgEntry 11 }

      tripSendReceiveMode OBJECT-TYPE
          SYNTAX INTEGER {
                  sendReceive(1),
                  sendOnly(2),
                  receiveOnly(3)
                  }
          MAX-ACCESS  read-only
          STATUS      current
          DESCRIPTION
              "The operational mode of this peer."
          ::= { tripCfgEntry 12 }

   --
   -- tripSupportedCommunityTable
   --

      tripSupportedCommunityTable   OBJECT-TYPE
          SYNTAX      SEQUENCE OF TripSupportedCommunityEntry
          MAX-ACCESS  not-accessible
          STATUS      current
          DESCRIPTION
              "The list of communities thatthis LS supports."
          ::= { trip 2 }

      tripSupportedCommunityEntry OBJECT-TYPE
          SYNTAX      TripSupportedCommunityEntry
          MAX-ACCESS  not-accessible
          STATUS      current
          DESCRIPTION
              "Entry containing information a community."
```

```
        INDEX { applIndex, tripSupportedCommunityId }
        ::= { tripSupportedCommunityTable 1 }

    TripSupportedCommunityEntry ::= SEQUENCE {
        tripSupportedCommunityId         TripItad,
        tripSupportedCommunityItad       TripItad,
        tripSupportedCommunityRowStatus  RowStatus
    }

    tripSupportedCommunityId OBJECT-TYPE
        SYNTAX      TripItad
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The identifier of the supported Community."
```



```
        ::= { tripSupportedCommunityEntry 1 }

    tripSupportedCommunityItad OBJECT-TYPE
        SYNTAX      TripItad
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "The Itad of the community."
        ::= { tripSupportedCommunityEntry 2 }

    tripSupportedCommunityRowStatus OBJECT-TYPE
        SYNTAX      RowStatus
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "The row status of the entry. This object is required to
            create or delete rows remotely by a manager."
        ::= { tripSupportedCommunityEntry 3 }

--
-- TripAuthMechanismTable
--

    tripAuthMechanismTable   OBJECT-TYPE
        SYNTAX      SEQUENCE OF TripAuthMechanismEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The list of authentication mechanisms and associated
            public keys that this LS supports."
        ::= { trip 3 }
```

```
    tripAuthMechanismEntry OBJECT-TYPE
        SYNTAX      TripAuthMechanismEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "Entry containing information about the connection with
            a TRIP peer."
        INDEX { applIndex, tripAuthMechanism }
        ::= { tripAuthMechanismTable 1 }

    TripAuthMechanismEntry ::= SEQUENCE {
        tripAuthMechanism              INTEGER,
        tripAuthMechPublicKey          TripPublicKey
    }

    tripAuthMechanism OBJECT-TYPE
        SYNTAX      INTEGER {
--          0 reserved
            dsa(1),
            rsa(2)
--          3 to 247 available
```



```
--          248, 249 for private use
--          250 to 255 reserved
            }
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The method used to compute authentication data.
            1 - DSA
            2 - RSA
            3 to 247 - available.
            248, 249 - for private use.
            0, 250 to 255 - reserved."
        ::= { tripAuthMechanismEntry 1 }

    tripAuthMechPublicKey OBJECT-TYPE
        SYNTAX      TripPublicKey
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The public key associated with this authentication
            mechanism."
        ::= { tripAuthMechanismEntry 2 }
```

```
   --
   -- 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 4 }

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

      TripPeerEntry ::= SEQUENCE {
          tripPeerRemoteAddrInetType           InetAddressType,
          tripPeerRemoteAddr                   InetAddress,
          tripPeerIdentifier                   TripId,
```



```
          tripPeerState                        INTEGER,
          tripPeerAdminStatus                  INTEGER,
          tripPeerNegotiatedVersion            Integer32,
          tripPeerSendReceiveMode              INTEGER,
          tripPeerSupportedProtocol            TripAppProtocol,
          tripPeerAddressFamily                TripAddressFamily,
          tripPeerRemotePort                   Integer32,
          tripPeerRemoteItad                   TripItad,
          tripPeerConnectRetryInterval         Integer32,
          tripPeerMaxRetryInterval             Integer32,
          tripPeerHoldTime                     Integer32,
          tripPeerKeepAlive                    Integer32,
          tripPeerHoldTimeConfigured           Integer32,
          tripPeerKeepAliveConfigured          Integer32,
          tripPeerMinItadOriginationInterval   Integer32,
          tripPeerMinRouteAdvertisementInterval Integer32,
```

```
        tripPeerMaxPurgeTime                    Integer32,
        tripPeerDisableTime                     Integer32,
        tripPeerAuthMechanism                   Integer32,
        tripPeerPublicKey                       TripPublicKey,
        tripPeerRowStatus                       RowStatus
    }


    tripPeerRemoteAddrInetType OBJECT-TYPE
        SYNTAX      InetAddressType
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The type of Inet Address of the tripPeerRemoteAddr."
        REFERENCE
            "RFC 2851, section 3."
        ::= { tripPeerEntry 1 }

    tripPeerRemoteAddr OBJECT-TYPE
        SYNTAX      InetAddress (SIZE(0..128))
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The remote IP address of this entry's TRIP peer. The
            size value of 128 has been assigned due to the sub
            identifier of object types length limitation as
            defined in SMIv2."
        REFERENCE
            "RFC 2851, section 3."
        ::= { tripPeerEntry 2 }

    tripPeerIdentifier OBJECT-TYPE
        SYNTAX      TripId
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "TRIP identifier of this entry's TRIP peer."
```



```
        ::= { tripPeerEntry 3 }

    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.

            idle(1)       : The initial state. Local LS refuses all
                            incoming connections. No resources are
                            allocated to the peer.

            connect(2)    : Local LS waiting for a transport protocol
                            connection to be completed to the peer,
                            and is listening for inbound transport
                            connections from the peer.

            active(3)     : LS is listening for an inbound connection
                            from the peer, but is not in the process
                            of initiating a connection to the peer.

            openSent(4)   : LS has sent an OPEN message to its peer
                            and is waiting for an OPEN message from
                            its peer.

            openConfirm(5): LS has sent an OPEN to its peer, received
                            an OPEN from its peer, and sent a
                            KEEPALIVE in response to the OPEN. The LS
                            is now waiting for a KEEPALIVE or
                            NOTIFICATION message in response to its
                            OPEN.

            established(6): LS can exchange UPDATE, NOTIFICATION,and
                            KEEPALIVE messages with its peer."
        ::= { tripPeerEntry 4 }

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



```
        ::= { tripPeerEntry 5 }
```

```
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 6 }

tripPeerSendReceiveMode OBJECT-TYPE
    SYNTAX      INTEGER {
            sendReceive(1),
            sendOnly(2),
            receiveOnly(3)
            }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The operational mode of this peer."
    ::= { tripPeerEntry 7 }

tripPeerSupportedProtocol OBJECT-TYPE
    SYNTAX      TripAppProtocol
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The protocol that this peer is using."
    ::= { tripPeerEntry 8 }

tripPeerAddressFamily OBJECT-TYPE
    SYNTAX      TripAddressFamily
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The address family that this peer belongs."
    ::= { tripPeerEntry 9 }

tripPeerRemotePort OBJECT-TYPE
    SYNTAX      Integer32 (1..65535)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The remote port for the TCP connection between the
        TRIP peers."
    ::= { tripPeerEntry 10 }

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

```
          DESCRIPTION
```



```
              "The Internet Telephony Administrative domain of
              this peer."
          ::= { tripPeerEntry 11 }

      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      { 120 }
          ::= { tripPeerEntry 12 }

      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 13 }

      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 14 }

      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 15 }

    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 16 }

    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 17 }

    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 { 30 }
        ::= { tripPeerEntry 18 }

    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 19 }

        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."
            DEFVAL { 10 }
            ::= { tripPeerEntry 20 }
```



```
        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."
            DEFVAL { 180 }
            ::= { tripPeerEntry 21 }

        tripPeerAuthMechanism OBJECT-TYPE
            SYNTAX      Integer32 (0..255)
            MAX-ACCESS  read-create
            STATUS      current
            DESCRIPTION
                "The authentication mechanism used with this peer."
            ::= { tripPeerEntry 22 }

        tripPeerPublicKey OBJECT-TYPE
            SYNTAX      TripPublicKey
            MAX-ACCESS  read-create
            STATUS      current
            DESCRIPTION
                "The public key for authentication of attributes from
```

```
            a peer."
        ::= { tripPeerEntry 23 }

    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 24 }

--
-- TripPeerStatsTable
--
    tripPeerStatsTable   OBJECT-TYPE
        SYNTAX      SEQUENCE OF TripPeerStatsEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The TRIP peer stats table. This table contains one entry
            per TRIP peer, and statistics related to the connection
            with the peer."
        ::= { trip 5 }
```



```
    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 6 }

      tripRouteEntry OBJECT-TYPE
          SYNTAX      TripRouteEntry
          MAX-ACCESS  not-accessible
          STATUS      current
          DESCRIPTION
              "Information about a route to a called destination."
          INDEX { applIndex,
                  tripRouteAppProtocol,
                  tripRouteAddressFamily,
                  tripRouteAddress,
                  tripRoutePeer
                  }
          ::= { tripRouteTable 1 }

      TripRouteEntry ::= SEQUENCE {
```

```
                tripRouteAppProtocol                TripAppProtocol,
                tripRouteAddressFamily              TripAddressFamily,
                tripRouteAddress                    OCTET STRING,
                tripRoutePeer                       TripId,
                tripRouteAddressSequenceNumber      Integer32,
                tripRouteAddressOriginatorId        TripItad,
                tripRouteNextHopServerIAddrType     InetAddressType,
                tripRouteNextHopServer              InetAddress,
                tripRouteNextHopServerPort          Integer32,
                tripRouteNextHopServerItad          TripItad,
                tripRouteMultiExitDisc              Unsigned32,
                tripRouteLocalPref                  Unsigned32,
                tripRouteAdvertisementPathSegment   OCTET STRING,
                tripRouteRoutedPathSegment          OCTET STRING,
                tripRouteAtomicAggregate            TruthValue,
                tripRouteBest                       TruthValue,
                tripRouteUnknown                    OCTET STRING,
                tripRouteWithdrawn                  TruthValue,
                tripRouteConverted                  TruthValue,
                tripRouteCircuitCapacity            Unsigned32,
                tripRouteDSPCapacity                Unsigned32
            }

    tripRouteAppProtocol OBJECT-TYPE
        SYNTAX      TripAppProtocol
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The protocol for which this routing table is
            maintained."
        ::= { tripRouteEntry 1 }

    tripRouteAddressFamily OBJECT-TYPE
        SYNTAX      TripAddressFamily
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "Specifies the type of address for the destination
            route."
        ::= { tripRouteEntry 2 }

    tripRouteAddress OBJECT-TYPE
        SYNTAX      OCTET STRING (SIZE(1..32))
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "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 }

      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 5 }

      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 6 }

      tripRouteNextHopServerIAddrType OBJECT-TYPE
          SYNTAX      InetAddressType
          MAX-ACCESS  read-only
          STATUS      current
          DESCRIPTION
              "The type of Inet Address of the tripRouteNextHopServer."
          REFERENCE
              "RFC 2851, section 3."
          ::= { tripRouteEntry 7 }

      tripRouteNextHopServer OBJECT-TYPE
          SYNTAX      InetAddress
          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 8 }

        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 9 }

        tripRouteNextHopServerItad OBJECT-TYPE
            SYNTAX      TripItad
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                "Indicates the domain of the next hop."
            ::= { tripRouteEntry 10 }

        tripRouteMultiExitDisc OBJECT-TYPE
            SYNTAX      Unsigned32 (0..4294967295)
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                "Used to descriminate between multiple exit points to
                an adjacent ITAD."
            ::= { tripRouteEntry 11 }

        tripRouteLocalPref OBJECT-TYPE
            SYNTAX      Unsigned32 (0..4294967295)
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                "Indicated the LS's degree of preference for an advertised
                route destination."
            ::= { tripRouteEntry 12 }

        tripRouteAdvertisementPathSegment OBJECT-TYPE
            SYNTAX      OCTET STRING (SIZE(4..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. This object is a sequence
                of ITADs in network byte order."
            ::= { tripRouteEntry 13 }

        tripRouteRoutedPathSegment OBJECT-TYPE
            SYNTAX      OCTET STRING (SIZE(4..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.  This object is a sequence
```



```
                of ITADs in network byte order."
            ::= { tripRouteEntry 14 }

        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 15 }

        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 16 }

        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 17 }

    tripRouteWithdrawn OBJECT-TYPE
        SYNTAX      TruthValue
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "Indicates if this route has been withdrawn but still
            maintained int the database less than the purge time."
        ::= { tripRouteEntry 18 }

    tripRouteConverted OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "Indicates if this route has been converted to a
            different application protocol than it had origionally."
        ::= { tripRouteEntry 19 }

    tripRouteCircuitCapacity OBJECT-TYPE
        SYNTAX      Unsigned32 (0..4294967295)
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
```



```
            "Represents the number of circuits remaining for
            terminating calls to this route."
        ::= { tripRouteEntry 20 }

    tripRouteDSPCapacity OBJECT-TYPE
        SYNTAX      Unsigned32 (0..4294967295)
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "Represents the number of MIPS remaining for terminating
            calls to this route."
        ::= { tripRouteEntry 21 }

   --
   -- TRIP Received Route CommunityTable.
   --

    tripRouteCommunityTable OBJECT-TYPE
```

```
        SYNTAX      SEQUENCE OF TripRouteCommunityEntry
        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 7 }

    tripRouteCommunityEntry OBJECT-TYPE
        SYNTAX      TripRouteCommunityEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "Information about a routes communities. An entry with
            a tripRouteAddress of 00 and a tripRoutePeer of 0 refers
            to the local LS."
        INDEX { applIndex,
                tripRouteAppProtocol,
                tripRouteAddressFamily,
                tripRouteAddress,
                tripRoutePeer,
                tripRouteCommunityId
                }
        ::= { tripRouteCommunityTable 1 }

    TripRouteCommunityEntry ::= SEQUENCE {
         tripRouteCommunityId    TripItad,
         tripRouteCommunityItad  TripItad
         }

    tripRouteCommunityId OBJECT-TYPE
        SYNTAX      TripItad
        MAX-ACCESS  not-accessible
        STATUS      current
```



```
        DESCRIPTION
            "The community identifier."
        ::= { tripRouteCommunityEntry 1 }

    tripRouteCommunityItad OBJECT-TYPE
        SYNTAX      TripItad
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The ITAD associated with this community."
```

```
        ::= { tripRouteCommunityEntry 2 }

   --
   -- 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 8 }

      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 9 }

    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
   -- *************

      tripNotifications   OBJECT IDENTIFIER ::= { trip 0 }

      tripEstablished NOTIFICATION-TYPE
          STATUS  current
          DESCRIPTION
              "The TRIP Established event is generated when the TRIP
              FSM enters the ESTABLISHED state."
          ::= { tripNotifications 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."
          ::= { tripNotifications 2 }

      tripOpenMessageError NOTIFICATION-TYPE
          OBJECTS { tripPeerLastError,
                    tripPeerState
                  }
          STATUS  current
          DESCRIPTION
              "Errors detected while processing the OPEN message."
          ::= { tripNotifications 3 }

      tripUpdateMessageError NOTIFICATION-TYPE
          OBJECTS { tripPeerLastError,
                    tripPeerState
                  }
          STATUS  current
          DESCRIPTION
              "Errors detected while processing the UPDATE message."
          ::= { tripNotifications 4 }

      tripHoldTimerExpired NOTIFICATION-TYPE
          OBJECTS { tripPeerLastError,
```

```
                  tripPeerState
                }
        STATUS  current
        DESCRIPTION
            "The system does not receive successive messages within
            the period specified by the negotiated Hold Time."
```



```
        ::= { tripNotifications 5 }

    tripConnectionCollision NOTIFICATION-TYPE
        STATUS  current
        DESCRIPTION
            "A pair of LSs tried to simultaneously to establish a
            transport connection to each other."
        ::= { tripNotifications 6 }


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

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

    tripPeerTableConfigGroup OBJECT-GROUP
        OBJECTS {
```

```
            tripPeerIdentifier,
            tripPeerState,
            tripPeerAdminStatus,
            tripPeerNegotiatedVersion,
            tripPeerSendReceiveMode,
            tripPeerSupportedProtocol,
            tripPeerAddressFamily,
            tripPeerRemotePort,
            tripPeerRemoteItad,
            tripPeerConnectRetryInterval,
            tripPeerMaxRetryInterval,
            tripPeerHoldTime,
            tripPeerKeepAlive,
            tripPeerHoldTimeConfigured,
```



```
            tripPeerKeepAliveConfigured,
            tripPeerMinItadOriginationInterval,
            tripPeerMinRouteAdvertisementInterval,
            tripPeerMaxPurgeTime,
            tripPeerDisableTime,
            tripPeerAuthMechanism,
            tripPeerPublicKey,
            tripPeerRowStatus
            }

        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 {
            tripRouteAddressSequenceNumber,
            tripRouteAddressOriginatorId,
            tripRouteNextHopServerIAddrType,
            tripRouteNextHopServer,
            tripRouteNextHopServerPort,
            tripRouteNextHopServerItad,
            tripRouteMultiExitDisc,
            tripRouteLocalPref,
            tripRouteAdvertisementPathSegment,
            tripRouteRoutedPathSegment,
            tripRouteAtomicAggregate,
            tripRouteBest,
            tripRouteUnknown,
            tripRouteWithdrawn,
            tripRouteConverted,
            tripRouteCircuitCapacity,
            tripRouteDSPCapacity,
            tripRouteCommunityItad
            }
```



```
        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,
            tripOpenMessageError,
            tripUpdateMessageError,
            tripHoldTimerExpired,
```

```
            tripConnectionCollision
        }
        STATUS  current
        DESCRIPTION
             "A collection of notifications defined for TRIP."
        ::= { tripGroups 6 }

END
```

7. 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 [23] and the View-based Access Control Model RFC 2575 [20] 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.

8. Revision History

8.1. Changes from <draft-zinman-trip-mib-00.txt>

- o Removed tripRouteAddressLen from the tripRouteTable.
- o Made use of INET-ADDRESS-MIB and it's TC's for IPv6 compliance.
- o Changed order of the enumeration of tripPeerSendReceiveMode to comform to TRIP draft.
- o Added objects tripPeerCircuitCapacity and tripPeerDSPCapacity to support TRIP for Gateways [24].
- o Removed tripPeerLastError and tripPeerState objects for tripEstablished notification.
- o Added local community object.
- o Added communities table for TRIP routes and removed community object from route table.
- o Added send/receive capability to local LS.
- o Added tripRouteAddressFamily as an Index to TripRouteEntry.
- o Changed enumerations in tripRouteAddressFamily to decimal(1) and hexaDecimal(2).
- o Support for authentication mechanism from draft-ietf-iptel-trip-authen-00.txt.
- o changed name of tripRoutePathSegment to tripRouteRoutedPathSegment.
- o Added tripRouteConverted to the routing table to signify a Converted Route.
- o Changed DEFVAL of tripPeerConnectRetryInterval from 60 to 120 seconds.
- o Added DEFVAL to tripPeerKeepAlive of 30 seconds.
- o Added DEFVAL to tripMaxPurgeTime and tripPeerMaxPurgeTime of 10 seconds.
- o Added DEFVAL to tripDisableTime and tripPeerDisableTime of 180 seconds.
- o Changed DEFVAL of tripMinItadOriginationInterval and tripPeerMinItadOriginationInterval to 30 seconds.
- o Removed tripHoldTimeConfigured and tripKeepAliveConfigured from TripCfgEntry.
- o changed names from opMode to sendReceiveMode.

9. References

1 Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

2 Rosenberg, J., Salama, H. and Squire, M., "Telephony Routing over IP (TRIP)", draft-ietf-iptel-trip-04.txt, work in progress.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

18 Case, J., McCloghrie, K., Rose, M. and Waldbusser, S., "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1905, January 1996.

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

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

21 Rosenberg, J. and Salama, H., "Authentication Attribute for TRIP", draft-ietf-iptel-trip-authen-00.txt, work in progress.

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

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

24 Rosenberg, J. and Salama, H., "Usage of TRIP in Gateways for Exporting Phone Routes", draft-rs-trip-gw-01.txt, work in progress.

10. Author's Address

Dave Walker
SS8 Networks Inc.
495 March Road
Ottawa, ON K2K 3G1
Canada
Phone: +1 613 592 2100
Email: drwalker@ss8.com

David Zinman
SS8 Networks Inc.
495 March Road

Ottawa, ON K2K 3G1

Canada
Phone: +1 613 592 2100
Email: david@ss8.com

Intellectual Property

Full Copyright Statement