D. Zinman
Internet Draft D. Walker
Document: draft-ietf-iptel-trip-mib-02.txt SS8 Networks
Expires: July 2002 Jianping Jiang
SS8 Networks

Management Information Base for Telephony Routing over IP (TRIP) <draft-ietf-iptel-trip-mib-02.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) [RFC3219] devices.

Since TRIP [RFC3219] is modeled after the Border Gateway Protocol (BGP-4) [RFC1771], the managed objects for TRIP are also modeled after RFC1657 - Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP-4) using SMIv2 [RFC2578].

Table of Contents

1. Introduction

This memo defines a portion of the Management Information Base (MIB)for use with network management protocols in the Internet community. In particular, it describes 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 BCP-0014 [BCP0014].

3. The SNMP Management Framework

The SNMP Management Framework presently consists of five major components:

- An overall architecture, described in RFC 2571 [RFC2571].
- 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 [RFC1155], STD 16, RFC 1212 [RFC1212] and RFC 1215 [RFC1215]. The second version, called SMIv2, is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

- Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in STD 15, RFC 1157 [RFC1157]. A second version of the SNMP message protocol, which is not an Internet standards

track protocol, is called SNMPv2c and described in RFC 1901 [RFC1901] and RFC 1906 [RFC1906]. The third version of the message protocol is called SNMPv3 and described in RFC 1906 [RFC1906], RFC 2572 [RFC2572] and RFC 2574 [RFC2574].

- Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in STD 15, RFC 1157 [RFC1157]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [RFC1905].

- A set of fundamental applications described in RFC 2573 [RFC2573] and the view-based access control mechanism described in RFC 2575 [RFC2575].

A more detailed introduction to the current SNMP Management Framework can be found in RFC 2570 [RFC2570].

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) [RFC3219] is an inter-domain application-
layer control protocol that exchanges information between TRIP location servers to provide efficient IP telephony routing. This MIB provides managed objects for TRIP devices defined in Telephony Routing over IP [RFC3219].

5. Structure of TRIP MIB

This MIB utilizes the framework described in RFC 2788 [RFC2788] 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 LS in the system. Each TRIP LS will then have an applIndex associated with it. The value assigned to applIndex will represent the distinct instance of TRIP.

The TRIP MIB contains the following groups of objects:

o The tripConfigGroup contains the common configuration objects

applicable to all TRIP applications referenced by the applIndex.

o The tripPeerTableConfigGroup contains the configuration objects applicable to all TRIP peers of the LS referened by the applIndex.
o The tripRouteGroup contains the configuration objects related to the routes of all TRIBs of this LS.
o The tripItadTopologyGroup contains information about the topology of the TRIP ITADs concerning this LS.
o The tripPeerTableStatsGroup contains the statistical objects applicable to all TRIP peers of the LS referened by the applIndex.
o The tripNotificationGroup contains notifications that the TRIP application can generate.
o The tripNotifObjectGroup contains the objects needed by one or more of the notifications.

5.1 Textual Conventions

The data types TripItad and TripId are used as textual conventions in this document. A TRIP ITAD is described in [RFC3219]. A TRIP ID is used
as a distinct identifier for a TRIP LS. A TripAppProtocol is used to identify an application protocol. A TripAddressFamily is used to define an address family. Both TripAppProtocol and TripAddressFamily are OBJECT IDENTIFIERS and as such, a MIB implementor can define a private object of this type of textual convention. 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,
       OBJECT-IDENTITY,
       Unsigned32,
       Integer32,
       Counter32,
       mib-2
           FROM SNMPv2-SMI

       TEXTUAL-CONVENTION,
       DateAndTime,
       TimeInterval,
       TruthValue,
       TimeStamp,
       StorageType,
       RowStatus
```



```
           FROM SNMPv2-TC

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

       InetAddressType,
       InetAddress
           FROM INET-ADDRESS-MIB

       applIndex
           FROM NETWORK-SERVICES-MIB;

   tripMIB MODULE-IDENTITY
      LAST-UPDATED "200202280000Z" -- Feb 28, 2002
      ORGANIZATION "IETF IPTel Working Group"
      CONTACT-INFO
      "Co-editor  David Zinman
       email:     dzinman@sympatico.ca
       phone:     +1 613 791 2841

       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  Jianping Jiang
```

```
                SS8 Networks, Inc.
        postal:    495 March Road, Suite #500
                   Ottawa, ON, K2K 3G1 Canada
        email:     jianping@ss8.com
        phone:     +1 613 592 2100
        "
    DESCRIPTION
        "The MIB module describing Telephony Routing over IP (TRIP).
        TRIP is a policy driven inter-administrative domain protocol
        for advertising the reachability of telephony destinations
        between location servers (LS), and for advertising attributes
        of the routes to those destinations."
    REVISION      "200202280000Z"
    DESCRIPTION
        "The initial version, Published as RFC xxxx."
    ::= { mib-2 xxx } -- to be assigned by IANA

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



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

    TripId ::= TEXTUAL-CONVENTION
        STATUS current
        DESCRIPTION
           "The TRIP Identifier uniquely identifies a LS within its
           ITAD. It is a 4 octet unsigned integer that may, but not
           necessarily, represent the IPv4 address of a Location
           Server (LS). For an IPv6 network, it will not represent the
           IPv6 address."
        SYNTAX Unsigned32 (0..4294967295)

    TripAppProtocol ::= TEXTUAL-CONVENTION
        STATUS current
        DESCRIPTION
            "The application protocol used for communication with TRIP
            Location Servers (LS). Protocols defined in this document
            are:
                tripSupProtSIP
                tripSupProtH323Q931
                tripSupProtH323RAS
```

```
                tripSupProtH323ANNEXG

            Users can add their own application protocol types by
            defining a TripAppProtocol type in a private
            specification."
        SYNTAX OBJECT IDENTIFIER

    TripAddressFamily ::= TEXTUAL-CONVENTION
        STATUS current
        DESCRIPTION
            "A type of address for a TRIP route. Address families
            defined within this MIB module are:
                tripAddrFamilyDecimal
                tripAddrFamilyPentadecimal
                tripAddrFamilyE164

           Users can add their own address family types by defining a
           TripAddressFamily type in a private specification."
        SYNTAX OBJECT IDENTIFIER

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

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



```
        DESCRIPTION
           "The version number of the TRIP protocol."
        SYNTAX Integer32 (1..255)

    TripSendReceiveMode ::= TEXTUAL-CONVENTION
        STATUS current
        DESCRIPTION
           "The operational mode of the TRIP application."
        SYNTAX INTEGER { sendReceive(1), sendOnly(2), receiveOnly(3) }

        tripMIBNotifications OBJECT IDENTIFIER ::= { tripMIB 0 }
        tripMIBObjects       OBJECT IDENTIFIER ::= { tripMIB 1 }
        tripMIBConformance   OBJECT IDENTIFIER ::= { tripMIB 2 }
        tripMIBNotifObjects  OBJECT IDENTIFIER ::= { tripMIB 3 }
        tripMIBAdmin         OBJECT IDENTIFIER ::= { tripMIB 4 }

        tripMIBCompliance    OBJECT IDENTIFIER ::=
                                    { tripMIBConformance 1 }
        tripMIBGroups        OBJECT IDENTIFIER ::=
```

```
                                    { tripMIBConformance 2 }

   --
   -- Supported protocols
   --
       tripSupportedProtocols OBJECT-IDENTITY
          STATUS        current
          DESCRIPTION
              "Registration point for the protocols supported by
              TRIP."
          ::= { tripMIBAdmin 1 }

       tripSupProtSIP
           OBJECT IDENTIFIER ::= { tripSupportedProtocols 1 }
       tripSupProtH323Q931
           OBJECT IDENTIFIER ::= { tripSupportedProtocols 2 }
       tripSupProtH323RAS
           OBJECT IDENTIFIER ::= { tripSupportedProtocols 3 }
       tripSupProtH323ANNEXG
           OBJECT IDENTIFIER ::= { tripSupportedProtocols 4 }

   --
   -- Address Families
   --
       tripAddressFamilies OBJECT-IDENTITY
          STATUS        current
          DESCRIPTION
              "Registration point for the address families supported
              by TRIP."
          ::= { tripMIBAdmin 2 }

       tripAddrFamilyDecimal
           OBJECT IDENTIFIER ::= { tripAddressFamilies 1 }

Zinman/Walker/Jiang                                                 7

Internet Draft                                          February 2002

       tripAddrFamilyPentadecimal
           OBJECT IDENTIFIER ::= { tripAddressFamilies 2 }
       tripAddrFamilyE164
           OBJECT IDENTIFIER ::= { tripAddressFamilies 3 }

   --
   -- tripCfgTable
   --
      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 applications referenced by the
             applIndex.  Each row represents those objects for a
             particular TRIP LS present in this system. The
             instances of TRIP LS's are uniquely identified by the
             applIndex."
        ::= { tripMIBObjects 1 }

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

    TripCfgEntry ::=
        SEQUENCE {
           tripCfgProtocolVersion              TripProtocolVersion,
           tripCfgItad                         TripItad,
           tripCfgIdentifier                   TripId,
           tripCfgOperStatus                   INTEGER,
           tripCfgAdminStatus                  INTEGER,
           tripCfgAddrIAddrType                InetAddressType,
           tripCfgAddr                         InetAddress,
           tripCfgPort                         Integer32,
           tripCfgMinItadOriginationInterval   Integer32,
           tripCfgMinRouteAdvertisementInterval Integer32,
           tripCfgMaxPurgeTime                 Integer32,
           tripCfgDisableTime                  Integer32,
           tripCfgSendReceiveMode              INTEGER
        }

    tripCfgProtocolVersion    OBJECT-TYPE
        SYNTAX     TripProtocolVersion
        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 the TRIP specification[RFC3219]."
        ::= { tripCfgEntry 1 }
```

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

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

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

         up(1)  : Set the application to normal operation.

         down(2): Set the application to a state where it will
                  not process TRIP messages."
    ::= { tripCfgEntry 4 }

tripCfgOperStatus OBJECT-TYPE
    SYNTAX      INTEGER {
                    up(1),
                    down(2)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The current operational state of the TRIP protocol.

         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 due to a fault or if
                    TRIP is in an initialization state.

            If tripAdminStatus is down(2) then tripOperStatus should
            be down(2). If tripAdminStatus is changed to up(1) then
            tripOperStatus should change to up(1) if there is no
            fault that prevents the TRIP protocol from moving to the
            up(1) state."
        ::= { tripCfgEntry 5 }

    tripCfgAddrIAddrType OBJECT-TYPE
        SYNTAX      InetAddressType
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The type of Inet Address of the tripAddr."
        REFERENCE
            "RFC 2851, section 3."
        ::= { tripCfgEntry 6 }

    tripCfgAddr OBJECT-TYPE
        SYNTAX      InetAddress
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The IP address of the local LS that the peer connects
            to."
        REFERENCE
            "RFC 2851, section 3."
        ::= { tripCfgEntry 7 }

    tripCfgPort OBJECT-TYPE
        SYNTAX      Integer32 (1..65535)
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "The local tcp/udp port on the local LS that the peer
            connects to."
        ::= { tripCfgEntry 8 }

    tripCfgMinItadOriginationInterval OBJECT-TYPE
        SYNTAX      Integer32 (1..2147483647)
        UNITS       "Seconds"
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "The minimum amount of time that must elapse between
            advertisement of update message that report changes
            within the LS's own ITAD."
        DEFVAL { 30 }
```

```
          ::= { tripCfgEntry 9 }

      tripCfgMinRouteAdvertisementInterval OBJECT-TYPE
          SYNTAX      Integer32 (1..2147483647)
          UNITS       "Seconds"
          MAX-ACCESS  read-write
          STATUS      current
          DESCRIPTION
              "Specifies minimal interval between successive
              advertisements to a particular destination from an LS."
          DEFVAL { 30 }
          ::= { tripCfgEntry 10 }

      tripCfgMaxPurgeTime OBJECT-TYPE
          SYNTAX      Integer32 (1..2147483647)
          UNITS       "Seconds"
          MAX-ACCESS  read-write
          STATUS      current
          DESCRIPTION
              "Indicate the interval that the LS must maintain routes
              marked as withdrawn in its database."
          DEFVAL { 10 }
          ::= { tripCfgEntry 11 }

      tripCfgDisableTime OBJECT-TYPE
          SYNTAX      Integer32 (1..2147483647)
          UNITS       "Seconds"
          MAX-ACCESS  read-write
          STATUS      current
          DESCRIPTION
              "Indicate the interval that the TRIP module of the
              LS must be disabled while routes originated by this
              location server with high sequence numbers can be
              removed."
          DEFVAL { 180 }
          ::= { tripCfgEntry 12 }

      tripCfgSendReceiveMode OBJECT-TYPE
          SYNTAX INTEGER {
                  sendReceive(1),
                  sendOnly(2),
                  receiveOnly(3)
                  }
          MAX-ACCESS  read-only
          STATUS      current
          DESCRIPTION
              "The operational mode of the trip entity running on this
```

```
             system."
          ::= { tripCfgEntry 13 }

   --
   -- TripRouteTypeTable
```



```
   --

       tripRouteTypeTable OBJECT-TYPE
         SYNTAX      SEQUENCE OF TripRouteTypeEntry
         MAX-ACCESS  not-accessible
         STATUS      current
         DESCRIPTION
             "The TRIP Route Type table contains one entry per
             supported protocol - address family pair. This table
             lists the route types supported in this peering session
             by the transmitting LS"
         ::= { tripMIBObjects 2 }

       tripRouteTypeEntry OBJECT-TYPE
         SYNTAX      TripRouteTypeEntry
         MAX-ACCESS  not-accessible
         STATUS      current
         DESCRIPTION
             "Entry containing information about the route type that
             the TRIP entity supports."
         INDEX { applIndex,
                 tripRouteTypeProtocolId,
                 tripRouteTypeAddrFamilyId }
           ::= { tripRouteTypeTable 1 }

       TripRouteTypeEntry ::= SEQUENCE {
           tripRouteTypeProtocolId         TripAppProtocol,
           tripRouteTypeAddrFamilyId       TripAddressFamily
       }

       tripRouteTypeProtocolId OBJECT-TYPE
         SYNTAX      TripAppProtocol
         MAX-ACCESS  not-accessible
         STATUS      current
         DESCRIPTION
             "The object identifier of a protocol that this peer is
             using."
         ::= { tripRouteTypeEntry 1 }

       tripRouteTypeAddrFamilyId OBJECT-TYPE
         SYNTAX      TripAddressFamily
```

```
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The object identifier of an address family that this
            peer belongs to."
        ::= { tripRouteTypeEntry 2 }

    --
    -- tripSupportedCommunityTable
    --
```



```
    tripSupportedCommunityTable   OBJECT-TYPE
        SYNTAX      SEQUENCE OF TripSupportedCommunityEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The list of TRIP communities that this LS supports. A
            TRIP community is a group of destinations that share
            common properties.

            The TRIP Communities attribute is used to group
            destinations so that the routing decision can be based
            on the identity of the group."
        REFERENCE
            "RFC 3219, section 5.9"
        ::= { tripMIBObjects 3 }

    tripSupportedCommunityEntry OBJECT-TYPE
        SYNTAX      TripSupportedCommunityEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "Entry containing information about a community. A TRIP
            community is a group of destinations that share some
            common property. This attribute is used so that routing
            decisions can be based on the identity of the group"
        INDEX { applIndex, tripSupportedCommunityId }
        ::= { tripSupportedCommunityTable 1 }

    TripSupportedCommunityEntry ::= SEQUENCE {
        tripSupportedCommunityId         TripCommunityId,
        tripSupportedCommunityItad       TripItad,
        tripSupportedCommunityStorage    StorageType,
        tripSupportedCommunityRowStatus  RowStatus
    }
```

```
      tripSupportedCommunityId OBJECT-TYPE
          SYNTAX      TripCommunityId
          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 }

     tripSupportedCommunityStorage OBJECT-TYPE
```



```
         SYNTAX       StorageType
         MAX-ACCESS   read-create
         STATUS       current
         DESCRIPTION
            "The storage type for this conceptual row.  Conceptual
            rows having the value 'permanent' need not allow write-
            access to any columnar objects in the row."
         DEFVAL { nonVolatile }
         ::= { tripSupportedCommunityEntry 3 }

      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 by a manager. A value for
              tripSupportedCommunityItad must be set for row creation
              to be successful. If the instance already exists for a
              particular applIndex, the row create operation will
              fail."
          ::= { tripSupportedCommunityEntry 4 }

   --
   -- 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."
        ::= { tripMIBObjects 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,
                tripPeerRemotePort }
          ::= {tripPeerTable 1}

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



```
        tripPeerRemotePort                  Integer32,
        tripPeerIdentifier                  TripId,
        tripPeerState                       INTEGER,
        tripPeerAdminStatus                 INTEGER,
        tripPeerNegotiatedVersion           TripProtocolVersion,
        tripPeerSendReceiveMode             INTEGER,
        tripPeerRemoteItad                  TripItad,
        tripPeerConnectRetryInterval        Integer32,
        tripPeerMaxRetryInterval            Integer32,
        tripPeerHoldTime                    Integer32,
        tripPeerKeepAlive                   Integer32,
        tripPeerHoldTimeConfigured          Integer32,
        tripPeerKeepAliveConfigured         Integer32,
        tripPeerMaxPurgeTime                Integer32,
        tripPeerDisableTime                 Integer32,
        tripPeerLearned                     TruthValue,
        tripPeerStorage                     StorageType,
        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..125))
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The remote IP address of this entry's TRIP peer. The
            size value of 125 has been assigned due to the sub
            identifier of object types length limitation as
            defined in SMIv2."
        REFERENCE
            "RFC 2851, section 3."
        ::= { tripPeerEntry 2 }

    tripPeerRemotePort OBJECT-TYPE
        SYNTAX      Integer32 (1..65535)
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The remote port for the TCP connection between the
            TRIP peers."
        ::= { tripPeerEntry 3 }
```



```
    tripPeerIdentifier OBJECT-TYPE
        SYNTAX      TripId
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "TRIP identifier of the peer."
        ::= { tripPeerEntry 4 }

    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 5 }

    tripPeerAdminStatus OBJECT-TYPE
```



```
        SYNTAX      INTEGER {
                        up(1),
                        down(2)
                    }
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "This object is used to affect the TRIP connection
            state.

            up(1)   : Allow a connection with the peer LS.
```

```
            down(2) : disconnect the connection from the peer LS and
                      do not allow any further connections to this
                      peer."
        ::= { tripPeerEntry 6 }

    tripPeerNegotiatedVersion OBJECT-TYPE
        SYNTAX      TripProtocolVersion
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The negotiated version of TRIP running between this
            local entity and this peer."
        ::= { tripPeerEntry 7 }

    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 8 }

    tripPeerRemoteItad OBJECT-TYPE
        SYNTAX      TripItad
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The Internet Telephony Administrative domain of
            this peer."
        ::= { tripPeerEntry 9 }

    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. This value must always be less
```

```
            than or equal to the value of tripPeerMaxRetryInterval.
            Attempts to set this value higher than the max retry
            should not be allowed."
        DEFVAL      { 120 }
        ::= { tripPeerEntry 10 }


    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. Once the value of
            tripPeerConnectRetryInterval has reached this value, no
            more retries will be attempted. Attempts to set this
            value lower than the retry interval should not be
            allowed."
        DEFVAL      { 360 }
        ::= { tripPeerEntry 11 }

    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 12 }

    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. This value is negotiated with the
            remote when a connection is established."
        ::= { tripPeerEntry 13 }
```

```
tripPeerHoldTimeConfigured OBJECT-TYPE
    SYNTAX      Integer32 (0 | 3..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 { 240 }
    ::= { tripPeerEntry 14 }

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 15 }

tripPeerMaxPurgeTime OBJECT-TYPE
    SYNTAX      Integer32 (1..65535)
    UNITS       "Seconds"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Indicate the interval that the LS must maintain routes
        marked as withdrawn in its database."
    DEFVAL { 10 }
    ::= { tripPeerEntry 16 }

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 peer
        LS must be disabled while routes originated by the local
        LS with high sequence numbers can be removed."
    DEFVAL { 180 }
    ::= { tripPeerEntry 17 }

 tripPeerLearned OBJECT-TYPE
     SYNTAX      TruthValue
     MAX-ACCESS  read-only
     STATUS      current
     DESCRIPTION
         "Indicates whether this entry was learned or
```

```
               configured."
```



```
           DEFVAL { false }
           ::= { tripPeerEntry 18 }

     tripPeerStorage OBJECT-TYPE
         SYNTAX       StorageType
         MAX-ACCESS   read-create
         STATUS       current
         DESCRIPTION
            "The storage type for this conceptual row.  Conceptual
            rows having the value 'permanent' need not allow write-
            access to any columnar objects in the row."
         DEFVAL { nonVolatile }
         ::= { tripPeerEntry 19 }

      tripPeerRowStatus 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. If that
              method already exists for a particular applIndex, the
              row create operation will fail."
          ::= { tripPeerEntry 20 }

   --
   --     tripPeerRouteTypeTable
   --
       tripPeerRouteTypeTable OBJECT-TYPE
         SYNTAX      SEQUENCE OF TripPeerRouteTypeEntry
         MAX-ACCESS  not-accessible
         STATUS      current
         DESCRIPTION
             "The TRIP peer Route Type table contains one entry per
             supported protocol - address family pair. Each instance
             of tripPeerRouteTypeEntry has an instance in the
             tripPeerTable as a parent."
         ::= { tripMIBObjects 5 }

       tripPeerRouteTypeEntry OBJECT-TYPE
         SYNTAX      TripPeerRouteTypeEntry
         MAX-ACCESS  not-accessible
         STATUS      current
         DESCRIPTION
             "Entry containing information about the route type that
```

```
            the TRIP peer supports."
          INDEX { applIndex,
                  tripPeerRemoteAddrInetType,
                  tripPeerRemoteAddr,
                  tripPeerRemotePort,
                  tripPeerRtTypeProtocolId,
                  tripPeerRtTypeAddrFamilyId }
```



```
          ::= { tripPeerRouteTypeTable 1 }

       TripPeerRouteTypeEntry ::= SEQUENCE {
           tripPeerRtTypeProtocolId         TripAppProtocol,
           tripPeerRtTypeAddrFamilyId       TripAddressFamily
       }

       tripPeerRtTypeProtocolId OBJECT-TYPE
         SYNTAX      TripAppProtocol
         MAX-ACCESS  not-accessible
         STATUS      current
         DESCRIPTION
             "The object identifier of a protocol that this peer is
             using."
         ::= { tripPeerRouteTypeEntry 1 }

       tripPeerRtTypeAddrFamilyId OBJECT-TYPE
         SYNTAX      TripAddressFamily
         MAX-ACCESS  read-only
         STATUS      current
         DESCRIPTION
             "The object identifier of an address family that this
             peer belongs to."
         ::= { tripPeerRouteTypeEntry 2 }

   --
   -- 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."
          ::= { tripMIBObjects 6 }

      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,
        tripPeerFsmEstablishedTransitions   Counter32,
        tripPeerFsmEstablishedTime          DateAndTime,
        tripPeerInUpdateElapsedTime         TimeInterval,
        tripPeerStateChangeTime             TimeStamp
    }

     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 LS."
        ::= { tripPeerStatsEntry 1 }

    tripPeerOutUpdates OBJECT-TYPE
        SYNTAX      Counter32
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The number of TRIP update messages sent 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 from the
            remote peer on this connection since the last restart
            of this LS."
```

```
        ::= { tripPeerStatsEntry 3 }

    tripPeerOutTotalMessages OBJECT-TYPE
        SYNTAX      Counter32
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The total number of outgoing TRIP messages sent to the
            remote peer since the last restart of this LS."
        ::= { tripPeerStatsEntry 4 }

    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
            LS."
```



```
        ::= { tripPeerStatsEntry 5 }

    tripPeerFsmEstablishedTime OBJECT-TYPE
        SYNTAX      DateAndTime
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "Indicates the time and date that this peer entered the
            established state."
        ::= { tripPeerStatsEntry 6 }

    tripPeerInUpdateElapsedTime OBJECT-TYPE
        SYNTAX      TimeInterval
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "Elapsed time in hundredths of seconds since the last
            TRIP update message was received from the peer."
        ::= { tripPeerStatsEntry 7 }

    tripPeerStateChangeTime OBJECT-TYPE
        SYNTAX       TimeStamp
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION
            "The value of sysUpTime when the last state change of
            tripPeerState took place."
```

```
        ::= { 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
              reachable routes that are to be added to service by the
              receiving LS."
          ::= { tripMIBObjects 7 }

      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,
          tripRouteTRIBMask                   BITS,
          tripRouteAddressSequenceNumber      Integer32,
          tripRouteAddressOriginatorId        TripId,
          tripRouteNextHopServerIAddrType     InetAddressType,
          tripRouteNextHopServer              InetAddress,
          tripRouteNextHopServerPort          Integer32,
          tripRouteNextHopServerItad          TripItad,
          tripRouteMultiExitDisc              Unsigned32,
          tripRouteLocalPref                  Unsigned32,
          tripRouteAdvertisementPath          OCTET STRING,
          tripRouteRoutedPath                 OCTET STRING,
```

```
        tripRouteAtomicAggregate            TruthValue,
        tripRouteUnknown                    OCTET STRING,
        tripRouteWithdrawn                  TruthValue,
        tripRouteConverted                  TruthValue,
        tripRouteReceivedTime               TimeStamp
        }

    tripRouteAppProtocol OBJECT-TYPE
        SYNTAX      TripAppProtocol
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The protocol for which this entry of the 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..255))
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "This is the address (prefix) of the family type given
```



```
            by Address Family of the destination. It is the prefix
            of addresses reachable from this gateway via the next
            hop server."
        REFERENCE
            "RFC 3219, section 5.1.1.1."
        ::= { 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 }
```

```
 tripRouteTRIBMask OBJECT-TYPE
    SYNTAX      BITS {
                 adjTribIns(0),
                 extTrib(1),
                 locTrib(2),
                 adjTribOut(3)
                 }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Indicates which TRIB(s) this entry belongs to. This is
        a bit-map of possible types. The various bit positions
        are:
        0    adjTribIns     The entry is of type adj-TRIBs-ins.
        1    extTrib        The entry is of type ext-TRIB.
        2    locTrib        The entry is of type loc-TRIB.
        3    adjTribOut     The entry is of type adj-TRIBs-out."
    DEFVAL { { } }
    ::= { tripRouteEntry 5 }

  tripRouteAddressSequenceNumber OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Indicates the version of the destination route
        originated by the LS identified by
        tripRouteAddressOriginatorId intra-domain attribute."
    ::= { tripRouteEntry 6 }

  tripRouteAddressOriginatorId OBJECT-TYPE
    SYNTAX      TripId
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This is an intra-domain attribute indicating the
```



```
        internal LS that originated the route into the ITAD."
    ::= { tripRouteEntry 7 }

  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 8 }

    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 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 (0..4294967295)
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This is used by an LS to express a preference for one
            link between the domains over another link between the
            domains.  A higher value represents a more preferred
            routing object."
        ::= { tripRouteEntry 12 }
```



```
    tripRouteLocalPref OBJECT-TYPE
        SYNTAX      Unsigned32 (0..4294967295)
        MAX-ACCESS  read-only
```

```
        STATUS      current
        DESCRIPTION
            "Indicated the local LS's degree of preference for an
            advertised route destination."
        ::= { tripRouteEntry 13 }

    tripRouteAdvertisementPath OBJECT-TYPE
        SYNTAX      OCTET STRING (SIZE(4..252))
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "Identifies the ITADs through wich routing information
            carried in an advertisement 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 14 }

    tripRouteRoutedPath OBJECT-TYPE
        SYNTAX      OCTET STRING (SIZE(4..252))
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "Identifies the ITADs through which messages sent using
            this route would pass. These are as subset of
            tripRouteAdvertisementPath.

            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 15 }

    tripRouteAtomicAggregate OBJECT-TYPE
        SYNTAX      TruthValue
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "Indicates that a route may traverse domains not listed
            in tripRouteRoutedPath. If an LS selects the less
            specific route from a set of overlapping routes, then
            this value returns TRUE."
        ::= { tripRouteEntry 16 }

    tripRouteUnknown OBJECT-TYPE
        SYNTAX      OCTET STRING (SIZE(0..255))
        MAX-ACCESS  read-only
```

```
          STATUS      current
          DESCRIPTION
              "One or more attributes not understood and dropped by
              this location server."
          REFERENCE
              "RFC 3219, section 4.3.2.3"
          ::= { tripRouteEntry 17 }

      tripRouteWithdrawn OBJECT-TYPE
          SYNTAX      TruthValue
          MAX-ACCESS  read-only
          STATUS      current
          DESCRIPTION
              "Indicates if this route is to be removed from service
              by the receiving LS."
          ::= { 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 originally."
          ::= { tripRouteEntry 19 }

      tripRouteReceivedTime OBJECT-TYPE
          SYNTAX       TimeStamp
          MAX-ACCESS   read-only
          STATUS       current
          DESCRIPTION
            "The value of sysUpTime when this route was received."
          ::= { tripRouteEntry 20 }

   --
   -- TRIP Received Route Community Table.
   --

      tripRouteCommunityTable OBJECT-TYPE
          SYNTAX      SEQUENCE OF TripRouteCommunityEntry
          MAX-ACCESS  not-accessible
          STATUS      current
          DESCRIPTION
              "A table containing a list of TRIP communities
              associated with a route. Each instance of
              tripPeerRouteTypeEntry has an instance in the
              tripRouteTable as a parent."
          REFERENCE
```

```
            "RFC 3219, section 5.9."
        ::= { tripMIBObjects 8 }

    tripRouteCommunityEntry OBJECT-TYPE
```



```
        SYNTAX      TripRouteCommunityEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "Information about communities associated with a route.
            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    TripCommunityId,
         tripRouteCommunityItad  TripItad
         }

    tripRouteCommunityId OBJECT-TYPE
        SYNTAX      TripCommunityId
        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."
          ::= { tripMIBObjects 9 }

      tripItadTopologyEntry OBJECT-TYPE
          SYNTAX      TripItadTopologyEntry
```



```
          MAX-ACCESS  not-accessible
          STATUS      current
          DESCRIPTION
              "Information about a peer of the LS identified by
              tripItadTopologyOrigId."
          INDEX { applIndex, tripItadTopologyOrigId }
          ::= { tripItadTopologyTable 1 }

      TripItadTopologyEntry ::= SEQUENCE {
                  tripItadTopologyOrigId    TripId,
                  tripItadTopologySeqNum    Unsigned32
              }

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

      tripItadTopologySeqNum OBJECT-TYPE
          SYNTAX      Unsigned32 (1..2147483647)
          MAX-ACCESS  read-only
          STATUS      current
          DESCRIPTION
              "Indicates the version of the ITAD topology
              originated by the LS identified by
              tripItadTopologyOrigId."
          ::= { tripItadTopologyEntry 2 }

   --
   -- tripItadTopologyIdTable
   --
```

```
      tripItadTopologyIdTable OBJECT-TYPE
          SYNTAX      SEQUENCE OF TripItadTopologyIdEntry
          MAX-ACCESS  not-accessible
          STATUS      current
          DESCRIPTION
              "The list of other LS's within the ITAD domain that the
              LS identified by tripItadTopologyOrigId is currently
              peering. Each instance of tripItadTopologyIdEntry has an
              instance in the tripItadTopologyTable as a parent."
          ::= { tripMIBObjects 10 }

      tripItadTopologyIdEntry OBJECT-TYPE
          SYNTAX      TripItadTopologyIdEntry
          MAX-ACCESS  not-accessible
          STATUS      current
          DESCRIPTION
```



```
              "Information about a peer to the LS identified by
              tripItadTopologyOrigId."
          INDEX { applIndex,
                  tripItadTopologyOrigId,
                  tripItadTopologyId }
          ::= { tripItadTopologyIdTable 1 }

      TripItadTopologyIdEntry ::= SEQUENCE {
                  tripItadTopologyId            TripId
              }

      tripItadTopologyId OBJECT-TYPE
          SYNTAX      TripId
          MAX-ACCESS  read-only
          STATUS      current
          DESCRIPTION
              "The index into this entry. Indicates the other location
              servers within the ITAD domain that this LS identified
              by tripItadTopologyOrigId is currently peering."
          ::= { tripItadTopologyIdEntry 1 }

   --
   -- Notification objects
   --

      tripNotifApplIndex    OBJECT-TYPE
          SYNTAX     Integer32 (1..2147483647)
          MAX-ACCESS accessible-for-notify
          STATUS     current
          DESCRIPTION
```

```
            "This object contains the applIndex as described
            in RFC 2788. It is used to bind this notification
            with a specific instance of TRIP entity."
        ::= { tripMIBNotifObjects 1 }

    tripNotifPeerAddrInetType OBJECT-TYPE
        SYNTAX      InetAddressType
        MAX-ACCESS  accessible-for-notify
        STATUS      current
        DESCRIPTION
            "The type of Inet Address of the tripNotifPeerAddr."
        REFERENCE
            "RFC 2851, section 3."
        ::= { tripMIBNotifObjects 2 }

    tripNotifPeerAddr OBJECT-TYPE
        SYNTAX      InetAddress (SIZE(0..125))
        MAX-ACCESS  accessible-for-notify
        STATUS      current
        DESCRIPTION
            "The remote IP address of this entry's TRIP peer. The
            size value of 125 has been assigned due to the sub
```



```
            identifier of object types length limitation as
            defined in SMIv2."
        REFERENCE
            "RFC 2851, section 3."
        ::= { tripMIBNotifObjects 3 }

    tripNotifPeerErrCode OBJECT-TYPE
        SYNTAX      INTEGER {
                        messageHeader(1),
                        openMessage(2),
                        updateMessage(3),
                        holdTimerExpired(4),
                        finiteStateMachine(5),
                        cease(6),
                        tripNotification(7)
                    }
        MAX-ACCESS  accessible-for-notify
        STATUS      current
        DESCRIPTION
            "Notification message of TRIP error. The meaning of this
            value is applicable to the following functions:

            messageHeader      - All errors detected while
                                 processing the TRIP message header.
```

```
            openMessage        - All errors detected while
                                 processing the OPEN message.
            updateMessage      - All errors detected while
                                 processing the UPDATE message.
            holdTimerExpired   - A notification generated when the
                                 hold timer expires.
            finiteStateMachine - All errors detected by the TRIP
                                 Finite State Machine.
            cease              - Any fatal error condition that the
                                 rest of the values do not cover.
            tripNotification   - Any error encountered while sending
                                 a notification message."
        ::= { tripMIBNotifObjects 4 }

    tripNotifPeerErrSubcode OBJECT-TYPE
        SYNTAX      Integer32 (1..2147483647)
        MAX-ACCESS  accessible-for-notify
        STATUS      current
        DESCRIPTION
            "The sub error code associated with error code. The
            meaning of this value is dependent on the value of
            tripNotifPeerErrCode.

            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.
            7 - Capability Mismatch.

            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."
        ::= { tripMIBNotifObjects 5 }

   --
```

```
   -- Notifications
   --
      tripEstablished NOTIFICATION-TYPE
          OBJECTS { tripNotifApplIndex }
          STATUS  current
          DESCRIPTION
              "The TRIP Established event is generated when the TRIP
              FSM enters the ESTABLISHED state."
          ::= { tripMIBNotifications 1 }

      tripFSM NOTIFICATION-TYPE
          OBJECTS { tripNotifApplIndex,
                    tripNotifPeerAddrInetType,
                    tripNotifPeerAddr,
                    tripNotifPeerErrCode,
                    tripNotifPeerErrSubcode,
                    tripPeerState
                  }
          STATUS  current
          DESCRIPTION
              "The trip FSM Event is generated when any error is
              detected by the TRIP Finite State Machine."
          ::= { tripMIBNotifications 2 }

      tripOpenMessageError NOTIFICATION-TYPE
          OBJECTS { tripNotifApplIndex,
                    tripNotifPeerAddrInetType,
                    tripNotifPeerAddr,
                    tripNotifPeerErrCode,
                    tripNotifPeerErrSubcode,
                    tripPeerState
                  }
          STATUS  current
          DESCRIPTION

Zinman/Walker/Jiang                                              33

Internet Draft                                        February 2002

              "Errors detected while processing the OPEN message."
          ::= { tripMIBNotifications 3 }

      tripUpdateMessageError NOTIFICATION-TYPE
          OBJECTS { tripNotifApplIndex,
                    tripNotifPeerAddrInetType,
                    tripNotifPeerAddr,
                    tripNotifPeerErrCode,
                    tripNotifPeerErrSubcode,
                    tripPeerState
                  }
          STATUS  current
```

```
        DESCRIPTION
            "Errors detected while processing the UPDATE message."
        ::= { tripMIBNotifications 4 }

    tripHoldTimerExpired NOTIFICATION-TYPE
        OBJECTS { tripNotifApplIndex,
                  tripNotifPeerAddrInetType,
                  tripNotifPeerAddr,
                  tripNotifPeerErrCode,
                  tripNotifPeerErrSubcode,
                  tripPeerState
                }
        STATUS  current
        DESCRIPTION
            "The system does not receive successive messages within
            the period specified by the negotiated Hold Time."
        ::= { tripMIBNotifications 5 }

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

    tripNotificationErr NOTIFICATION-TYPE
        OBJECTS { tripNotifApplIndex }
        STATUS  current
        DESCRIPTION
            "Generated if there is an error detected in a TRIP
            notification message sent with another cause. Note that
            the TRIP notification refered to in this object is not
            an SNMP notification, it is a specific message described
            in the TRIP specification."
        REFERENCE
            "RFC 3219, section  6.4."
        ::= { tripMIBNotifications 7 }

    --
```



```
    -- Compliance Statements
    --
    tripCompliance MODULE-COMPLIANCE
        STATUS     current
        DESCRIPTION
             "The compliance statement for TRIP entities."
```

```
        MODULE -- this module
             MANDATORY-GROUPS { tripConfigGroup,
                                tripPeerTableConfigGroup,
                                tripRouteGroup,
                                tripItadTopologyGroup,
                                tripPeerTableStatsGroup }

        GROUP tripNotificationGroup
        DESCRIPTION
            "This group is optional. A TRIP entity can choose not to
            send any notifications. If this group is implemented,
            the tripNotifObjectGroup must also be implemented."

        GROUP tripNotifObjectGroup
        DESCRIPTION
            "This group is optional. A TRIP entity can choose not to
            send any notifications. If this group is implemented,
            the tripNotificationGroup must also be implemented."

        MODULE NETWORK-SERVICES-MIB
             MANDATORY-GROUPS { applGroup }

        ::= { tripMIBCompliance 1 }

   --
   -- Object and event conformance groups
   --

      tripConfigGroup OBJECT-GROUP
          OBJECTS {
              tripCfgProtocolVersion,
              tripCfgItad,
              tripCfgIdentifier,
              tripCfgOperStatus,
              tripCfgAdminStatus,
              tripCfgAddrIAddrType,
              tripCfgAddr,
              tripCfgPort,
              tripCfgMinItadOriginationInterval,
              tripCfgMinRouteAdvertisementInterval,
              tripCfgMaxPurgeTime,
              tripCfgDisableTime,
              tripCfgSendReceiveMode,
              tripSupportedCommunityItad,
              tripSupportedCommunityStorage,
```



```
              tripRouteTypeAddrFamilyId,
```

```
        tripSupportedCommunityRowStatus
    }
    STATUS current
    DESCRIPTION
        "The global objects for configuring trip."
    ::= { tripMIBGroups 1 }

tripPeerTableConfigGroup OBJECT-GROUP
    OBJECTS {
        tripPeerIdentifier,
        tripPeerState,
        tripPeerAdminStatus,
        tripPeerNegotiatedVersion,
        tripPeerSendReceiveMode,
        tripPeerRemoteItad,
        tripPeerConnectRetryInterval,
        tripPeerMaxRetryInterval,
        tripPeerHoldTime,
        tripPeerKeepAlive,
        tripPeerHoldTimeConfigured,
        tripPeerKeepAliveConfigured,
        tripPeerMaxPurgeTime,
        tripPeerDisableTime,
        tripPeerLearned,
        tripPeerStorage,
        tripPeerRtTypeAddrFamilyId,
        tripPeerRowStatus
        }

    STATUS current
    DESCRIPTION
        "The global objects for configuring the TRIP peer
        table."
    ::= { tripMIBGroups 2 }

tripPeerTableStatsGroup OBJECT-GROUP
    OBJECTS {
        tripPeerInUpdates,
        tripPeerOutUpdates,
        tripPeerInTotalMessages,
        tripPeerOutTotalMessages,
        tripPeerFsmEstablishedTransitions,
        tripPeerFsmEstablishedTime,
        tripPeerInUpdateElapsedTime,
        tripPeerStateChangeTime
        }
    STATUS current
    DESCRIPTION
        "The global statistics the TRIP peer table."
    ::= { tripMIBGroups 3 }
```

```
      tripRouteGroup OBJECT-GROUP
          OBJECTS {
              tripRouteTRIBMask,
              tripRouteAddressSequenceNumber,
              tripRouteAddressOriginatorId,
              tripRouteNextHopServerIAddrType,
              tripRouteNextHopServer,
              tripRouteNextHopServerPort,
              tripRouteNextHopServerItad,
              tripRouteMultiExitDisc,
              tripRouteLocalPref,
              tripRouteAdvertisementPath,
              tripRouteRoutedPath,
              tripRouteAtomicAggregate,
              tripRouteUnknown,
              tripRouteWithdrawn,
              tripRouteConverted,
              tripRouteReceivedTime,
              tripRouteCommunityItad
              }

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

      tripItadTopologyGroup OBJECT-GROUP
          OBJECTS {
              tripItadTopologySeqNum,
              tripItadTopologyId
              }
          STATUS current
          DESCRIPTION
              "The objects that define the TRIP ITAD topology."
          ::= { tripMIBGroups 5 }

      tripNotificationGroup NOTIFICATION-GROUP
          NOTIFICATIONS {
              tripEstablished,
              tripFSM,
              tripOpenMessageError,
              tripUpdateMessageError,
              tripHoldTimerExpired,
              tripConnectionCollision,
              tripNotificationErr
          }
```

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

      tripNotifObjectGroup OBJECT-GROUP
```



```
          OBJECTS {
              tripNotifApplIndex,
              tripNotifPeerAddrInetType,
              tripNotifPeerAddr,
              tripNotifPeerErrCode,
              tripNotifPeerErrSubcode
              }
          STATUS current
          DESCRIPTION
              "The collection of objects that specify information for
              TRIP notifications."
          ::= { tripMIBGroups 7 }

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 [RFC2574] and the View-based Access Control Model RFC 2575 [RFC2575] 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>

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

TripCfgEntry.
- changed names from opMode to sendReceiveMode.

8.2. Changes from <draft-zinman-trip-mib-01.txt>

- Added tripOperStatus.
- Changed definition of textual convention TripAppProtocol to OBJECT IDENTIFIER. See [RFC3219] section 13.4
- Changed definition of textual convention TripAddressFamily to OBJECT IDENTIFIER. See [RFC3219] section 13.3
- Added object identifiers for tripSupportedProtocols and tripAddressFamilies.
- Removed authentication tables.
- Removed textual convention TripPublicKey.
- Changed the position of the MIB branches slightly.
- Changed name of tripPeerLastError to tripNotifPeerErrCode and MAX-ACCESS to accessible-for-notify.

- Separated tripNotifPeerErrSubcode from tripNotifPeerErrCode and made them both integers.
- Added compliance statements.
- Changed MAX-ACCESS of tripPeerRemoteItad to read-create.
- Changed DEFVAL of tripPeerHoldTimeConfigured from 90 to 240.

8.3. Changes from <draft-ietf-iptel-trip-mib-00.txt>

- Changed tripSupportedProtocols and tripAddressFamilies from OBJECT IDENTIFIER to OBJECT-IDENTITY.
- Added tripRouteTRIBMask with syntax BITS to identify the type of TRIB the route belongs to.
- Removed tripMinItadOriginationInterval and tripMinRouteAdvertisementInterval from the tripCfgTable because they also exist in the Peer table.
- TripPeerRemoteItad made read-only because either the local application will determine the value.
- Add tripRouteTypeTable as a sub-table to tripCfgTable (similar to tripPeerRouteTypeTable).
- Add timestamp to route table (when received), and last change of peer state.
- Removed tripRouteBest since the best would be represented by LocTRIB or AdjTRIBOut

8.4. Changes from <draft-ietf-iptel-trip-mib-01.txt>

- Reworded chapter 5 to reflect the use of NETWORK-SERVICES-MIB.
- Removed some rowStatus objects that were not needed.
- Added the branch tripMIBNotifObjects to hold the notification

objects.
- o Added the branch tripMIBAdmin to hold TRIP address families and TRIP supported protocols.
- o Document now references RFC 3219 for TRIP.
- o General cleanup of descriptions.
- o Made descriptors for consistent by using the first part of the table name as a prefix.
- o Added objects with the StorageType textual convention for read-create tables.
- o Updated the references section to conform to IETF specifications.
- o Added full copyright statement.

9. Full Copyright Statement

10. References

[RFC3219] Rosenberg, J., Salama, H. and Squire, M., "Telephony Routing over IP (TRIP)", RFC 3219 January 2002.

[RFC1771] Rekhter, Y. and Li, T., "Border Gateway Protocol 4 (BGP-4)", RFC 1771, March 1995.

[RFC2571] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", RFC 2571, April 1999.

[RFC1155] Rose, M., and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", STD 16, RFC 1155, May 1990.

[RFC1212] Rose, M., and K. McCloghrie, "Concise MIB Definitions", STD 16, RFC 1212, March 1991.

[RFC1215] M. Rose, "A Convention for Defining Traps for use with the SNMP", RFC 1215, March 1991.

[RFC2578] 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.

[RFC2579] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M., and S. Waldbusser, "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

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

[RFC1157] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol", STD 15, RFC 1157, May 1990.

[RFC1901] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Introduction to Community-based SNMPv2", RFC 1901, January 1996.

[RFC1906] 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.

[RFC2572] Case, J., Harrington D., Presuhn R., and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", RFC 2572, April 1999.

[RFC2574] Blumenthal, U., and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network

Management Protocol (SNMPv3)", RFC 2574, April 1999.

[RFC1905] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1905, January 1996.

[RFC2573] Levi, D., Meyer, P., and B. Stewart, "SNMPv3 Applications", RFC 2573, April 1999.

[RFC2575] Wijnen, B., Presuhn, R., and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", RFC 2575, April 1999.

[RFC2570] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction to Version 3 of the Internet-standard Network Management Framework", RFC 2570, April 1999.

[RFC2788] Freed, N., Kille, s., "Network Services Monitoring MIB", RFC 2788, March 2000.

[BCP0014] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

11. Author's Addresses

David Zinman
phone: +1 613 791 2841
Email: dzinman@sympatico.ca

Dave Walker
SS8 Networks, Inc.

495 March Road, Suite #500
Ottawa, ON K2K 3G1
Canada
Phone: +1 613 592 2100
Email: drwalker@ss8.com

Jianping Jiang
SS8 Networks, Inc.
495 March Road, Suite #500
Ottawa, ON K2K 3G1
Canada
Phone: +1 613 592 2100
Email: jianping@ss8.com

12. Working Group

IP Telephony (iptel) Working Group
==========

Mailing list: iptel@lists.bell-labs.com

Chair:
  Jonathan Rosenberg <jdrosen@dynamicsoft.com>

Transport Area Directors:
  Scott Bradner <sob@harvard.edu>
  Allison Mankin mankin@east.isi.edu