TEAS Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 18, 2016

V. Beeram
Juniper Networks
T. Saad
R. Gandhi
Cisco Systems Inc
X. Liu
Ericsson
H. Shah
Ciena
X. Chen
Huawei Technologies
R. Jones
Brocade
B. Wen
Comcast

October 16, 2015

A YANG Data Model for Resource Reservation Protocol (RSVP)
draft-ietf-teas-yang-rsvp-01

Abstract

This document defines a YANG data model for the configuration and management of RSVP Protocol. The model defines generic RSVP protocol building blocks that can be augmented and used by other RSVP extension models such as RVSP extensions to Traffic-Engineering (RSVP-TE). The model covers the RSVP protocol configuration, operational state, remote procedural calls, and event notifications data.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 18, 2016.

Table of Contents

1. Introduction

YANG [RFC6020] is a data definition language that was introduced to define the contents of a conceptual data store that allows networked devices to be managed using NETCONF [RFC6241]. YANG is proving relevant beyond its initial confines, as bindings to other interfaces (e.g. ReST) and encoding other than XML (e.g. JSON) are being defined. Furthermore, YANG data models can be used as the basis of implementation for other interface, such as CLI and programmatic APIs.

This document defines a YANG data model that can be used to configure and manage the RSVP protocol. This model covers generic protocol building blocks that can be augmented and used by other RSVP extension models- such as extensions for signaling RSVP-TE packet (or other technology specific) Label Switched Paths (LSP)s.

1.1. Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in BCP 14, RFC 2119 [RFC2119].

1.2. Tree Diagram

A simplified graphical representation of the data model is presented in each section of the model. The following notations are used for the YANG model data tree representation.

```
   <status> <flags> <name> <opts> <type>

    <status> is one of:
      +  for current
      x  for deprecated
      o  for obsolete

    <flags> is one of:
      rw  for read-write configuration data
      ro  for read-only non-configuration data
      -x  for execution rpcs
      -n  for notifications

    <name> is the name of the node

   If the node is augmented into the tree from another module, its name
   is printed as <prefix>:<name>

    <opts> is one of:
      ? for an optional leaf or node
      ! for a presence container
      * for a leaf-list or list
      Brackets [<keys>] for a list's keys
      Curly braces {<condition>} for optional feature that make node
   conditional
      Colon : for marking case nodes
      Ellipses ("...") subtree contents not shown

      Parentheses enclose choice and case nodes, and case nodes are also
      marked with a colon (":").

    <type> is the name of the type for leafs and leaf-lists.
```

1.3. Prefixes in Data Node Names

In this document, names of data nodes and other data model objects are prefixed using the standard prefix associated with the corresponding YANG imported modules, as shown in Table 1.

Prefix	YANG module	Reference
yang	ietf-yang-types	[RFC6991]
inet	ietf-inet-types	[RFC6991]

Table 1: Prefixes and corresponding YANG modules

1.4. Open Issues and Next Steps

This document covers YANG models for data pertaining to the base RSVP, the generic RSVP-TE, and the packet RSVP-TE protocols. The current revision of this draft covers configuration and state data, but future revisions are expected to cover data for RPCs, and notifications.

1.4.1. Module Hierarchy

During discussions, some of the RSVP features were debated whether they should be present in the base RSVP model or in extension RSVP model (e.g. RSVP-TE model) especially that some features were defined in RSVP extension drafts for GMPLS or RSVP-TE states. For example, the RSVP Hello extensions defined in [RFC3209] with extensions to RSVP for TE states. However, RSVP Hellos extension can also apply to non RSVP-TE states, and some vendor implementations, allow it to be enabled independent of RSVP-TE features.

1.4.2. Model Data Organization

Throughout the model, the approach described in [I-D.openconfig-netmod-opstate] is adopted to represent data pertaining to configuration intended state, applied state and derived state data elements. Each container in the model hold a "config" and "state" sub-container. The "config" sub-container is used to represent the intended configurable parameters, and the state sub-container is used to represent both the applied configurable parameters and any derived state, such as counters or statistical information.

The decision to use this approach was made to better align with the MPLS consolidated model in [I-D.openconfig-mpls-consolidated-model], and maximize reusability of groupings defined in this document and allow for possible convergence between the two models.

1.4.3. State Data

Pure state data (for example, protocol derived data) can be modeled using two options:

- Contained inside the read-write container, under the "state" sub-container, as shown in Figure 2

- Contained inside a separate read-only container

The first option allows for reusing the same containers that hold configuration read-write data under a "config" sub-container, and by

adding the state data under the read-only "state" sub-container of the container. For ephemeral or purely derived states (e.g. RSVP sessions), and since in this case the state would hang off a read-write parent container, it will be possible to delete the parent container and removing such state.

The second option entails defining a new read-only parent container in the model (e.g. neighbors-state) that holds the data.

This revision of the draft adopts the first option. Further discussions on this topic are expected to close on the best choice to adopt.

2. Design Considerations

2.1. Base Model(s)

The base RSVP model covers core features with the minimal set of configuration parameters needed to operate them. Additional core RSVP parameter configuration(s) as well as extended RSVP feature(s) are covered in a separate RSVP extended model.

The RSVP-TE [RFC3209] and other traffic-engineering specific enhancements are also addressed in separate module(s). The generic and packet specific RSVP traffic-engineering models are augmentations to the RSVP base model and are discussed in this revision of the document in Section 3.3.

Currently, the RSVP-TE module is presented as part of this draft, and is mostly packet centric. It is expected that the RSVP-TE YANG model will move to a separate document in the next revision.

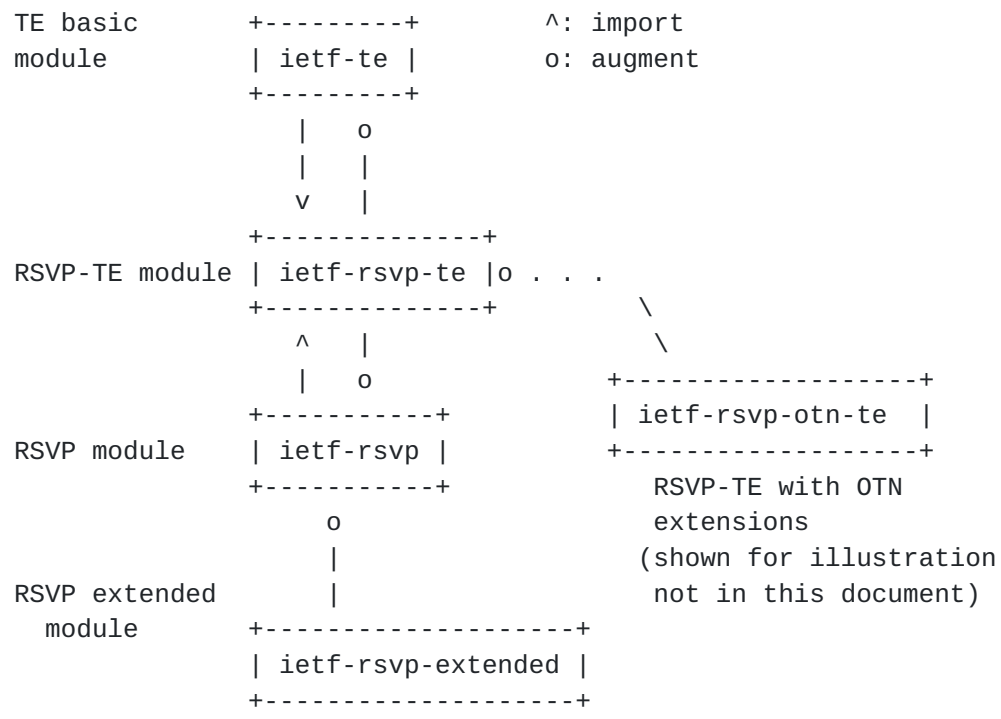

Figure 1: Relationship of RSVP and RSVP-TE modules with other protocol modules

The RSVP base model in this revision of the document does not aim to be feature complete. The primary intent is to cover a set of standard core features (listed below) that are commonly in use.

- Authentication ([RFC2747])
- Refresh Reduction ([RFC2961])
- Hellos ([RFC3209])
- Graceful Restart ([RFC3473], [RFC5063])

2.2. Extended Model(s)

The extended RSVP YANG model covers non-basic configuration(s) for RSVP core feature(s) as well as optional RSVP feature that are not a must for basic RSVP operation.

2.3. Configuration Inheritance

The defined data model supports configuration inheritance for neighbors, and interfaces. Data elements defined in the main container (e.g. the container that encompasses the list of interfaces, or neighbors) are assumed to apply equally to all

elements of the list, unless overridden explicitly for a certain element (e.g. interface). Vendors are expected to augment the above container(s) to provide the list of inheritance command for their implementations.

2.4. Vendor Configuration Models

There two main popular types of routing protocol configuration that vendors may support:

- protocol centric - all the protocol related configuration is contained within the protocol itself. Configuration belonging to multiple instances of the protocol running in different routing-instances (e.g. VRFs) are contained under the default routing instance [I-D.ietf-netmod-routing-cfg]:

- VRF centric - all the protocol related configuration for a routing- instance is contained within this routing-instance.

On-going discussions within the IETF community have converged on adopting the VRF centric approach. The proposed model in this document adheres to this conclusion.

3. Model Organization

This document defines YANG data models for RSVP base, RSVP extended, RSVP-TE generic, and RSVP-TE packet MPLS configuration, state, notification and RPCs. The relationship between the different modules is depicted in Figure 1.

3.1. RSVP Base YANG Model

This section describes the RSVP base YANG data model. It covers base RSVP protocol data defined by RSVP [RFC2205], and enhancements that pertain to the base protocol operation.

The container "rsvp" is the top level container in this data model. The presence of this container is expected to enable RSVP protocol functionality.

The approach described in [I-D.openconfig-netmod-opstate] allows for modeling the intended and respective applied and derived state. The TE state data in this model falls into one of the following categpries:

- State corresponding to applied configuration

- State corrsponding to derived state, counters, stats, etc.

Data for such state is contained under the respective "state" sub-container of the intended object (e.g. interface) as shown in Figure 2.

```
module: ietf-rsvp
   +--rw rsvp!
      +--rw globals
            +-- rw config
               <<intended configuration>>
            .
            +-- ro state
               <<applied configuration>>
               <<derived state associated with the tunnel>>
         .
         .
      +--rw interfaces
            +-- rw config
               <<intended configuration>>
            .
            +-- ro state
               <<applied configuration>>
               <<derived state associated with the tunnel>>
         .
         .
      +--rw neighbors
            +-- rw config
               <<intended configuration>>
            .
            +-- ro state
               <<applied configuration>>
               <<derived state associated with the tunnel>>
         .
         .
      +--rw sessions
            +-- rw config
               <<intended configuration>>
            .
            +-- ro state
               <<applied configuration>>
               <<derived state associated with the tunnel>>
         .
   rpcs:
      +--x global-rpc
      +--x interfaces-rpc
      +--x neighbors-rpc
      +--x sessions-rpc
   notifications:
      +--n global-notif
```

```
        +--n interfaces-notif
        +--n neighbors-notif
        +--n sessions-notif
```

Figure 2: RSVP highlevel model view

The following subsections provide overview of the parts of the model pertaining to configuration and state data.

3.1.1. Configuration and State Data

3.1.1.1. Global Data

This branch of the data model covers global configuration and states that control RSVP protocol behavior.

```
   module: ietf-rsvp
      +--rw rsvp!
         +--rw globals
         |  +--rw config
         |  |  +--rw graceful-restart
         |  |     +--rw enabled?   boolean
         |  +--ro state
         |     +--ro graceful-restart
         |     |  +--ro enabled?   boolean
         |     +--ro packets-stats
         |     |  +--ro sent?   yang:counter32
         |     |  +--ro rcvd?   yang:counter32
         |     +--ro protocol-stats
         |     |  +--ro ack-sent?                   yang:counter32
         |     |  +--ro ack-rcvd?                   yang:counter32
         |     |  +--ro bundle-sent?                yang:counter32
         |     |  +--ro bundle-rcvd?                yang:counter32
         |     |  +--ro hello-sent?                 yang:counter32
         |     |  +--ro hello-rcvd?                 yang:counter32
         |     |  +--ro integrity-challenge-sent?   yang:counter32
         |     |  +--ro integrity-challenge-rcvd?   yang:counter32
         |     |  +--ro integrity-response-sent?    yang:counter32
         |     |  +--ro integrity-response-rcvd?    yang:counter32
         |     |  +--ro notify-sent?                yang:counter32
         |     |  +--ro notify-rcvd?                yang:counter32
         |     |  +--ro path-sent?                  yang:counter32
         |     |  +--ro path-rcvd?                  yang:counter32
         |     |  +--ro path-err-sent?              yang:counter32
         |     |  +--ro path-err-rcvd?              yang:counter32
         |     |  +--ro path-tear-sent?             yang:counter32
         |     |  +--ro path-tear-rcvd?             yang:counter32
         |     |  +--ro resv-sent?                  yang:counter32
         |     |  +--ro resv-rcvd?                  yang:counter32
         |     |  +--ro resv-confirm-sent?          yang:counter32
         |     |  +--ro resv-confirm-rcvd?          yang:counter32
         |     |  +--ro resv-err-sent?              yang:counter32
         |     |  +--ro resv-err-rcvd?              yang:counter32
         |     |  +--ro resv-tear-sent?             yang:counter32
         |     |  +--ro resv-tear-rcvd?             yang:counter32
         |     |  +--ro summary-refresh-sent?       yang:counter32
         |     |  +--ro summary-refresh-rcvd?       yang:counter32
         |     |  +--ro unknown-recv?               yang:counter32
         |     +--ro errors-stats
         |        +--ro authenticate?   yang:counter64
         |        +--ro checksum?       yang:counter64
         |        +--ro packet-len?     yang:counter64
```

3.1.1.2. Interface Data

This branch of the data model covers configuration and state elements relevant to one or all RSVP interfaces. Any data configuration applied at the "interfaces" container level are equally applicable to all interfaces - unless overridden by explicit configuration under a specific interface.

```
module: ietf-rsvp
   +--rw rsvp!
      +--rw interfaces
      |  +--rw config
      |  |  +--rw refresh-reduction
      |  |  |  +--rw enabled?   boolean
      |  |  +--rw rsvp-hellos
      |  |  |  +--rw enabled?   boolean
      |  |  +--rw authentication
      |  |     +--rw enabled?     boolean
      |  |     +--rw password?    string
      |  |     +--rw algorithm?   identityref
      |  +--ro state
      |  |  +--ro refresh-reduction
      |  |  |  +--ro enabled?   boolean
      |  |  +--ro rsvp-hellos
      |  |  |  +--ro enabled?   boolean
      |  |  +--ro authentication
      |  |  |  +--ro enabled?     boolean
      |  |  |  +--ro password?    string
      |  |  |  +--ro algorithm?   identityref
      |  |  +--ro packets-stats
      |  |  |  +--ro sent?   yang:counter32
      |  |  |  +--ro rcvd?   yang:counter32
      |  |  +--ro protocol-stats
      |  |  |  +--ro ack-sent?                  yang:counter32
      |  |  |  +--ro ack-rcvd?                  yang:counter32
      |  |  |  +--ro bundle-sent?               yang:counter32
      |  |  |  +--ro bundle-rcvd?               yang:counter32
      |  |  |  +--ro hello-sent?                yang:counter32
      |  |  |  +--ro hello-rcvd?                yang:counter32
      |  |  |  +--ro integrity-challenge-sent?  yang:counter32
      |  |  |  +--ro integrity-challenge-rcvd?  yang:counter32
      |  |  |  +--ro integrity-response-sent?   yang:counter32
      |  |  |  +--ro integrity-response-rcvd?   yang:counter32
      |  |  |  +--ro notify-sent?               yang:counter32
      |  |  |  +--ro notify-rcvd?               yang:counter32
      |  |  |  +--ro path-sent?                 yang:counter32
      |  |  |  +--ro path-rcvd?                 yang:counter32
      |  |  |  +--ro path-err-sent?             yang:counter32
```

```
|  |  |  +--ro path-err-rcvd?              yang:counter32
|  |  |  +--ro path-tear-sent?             yang:counter32
|  |  |  +--ro path-tear-rcvd?             yang:counter32
|  |  |  +--ro resv-sent?                  yang:counter32
|  |  |  +--ro resv-rcvd?                  yang:counter32
|  |  |  +--ro resv-confirm-sent?          yang:counter32
|  |  |  +--ro resv-confirm-rcvd?          yang:counter32
|  |  |  +--ro resv-err-sent?              yang:counter32
|  |  |  +--ro resv-err-rcvd?              yang:counter32
|  |  |  +--ro resv-tear-sent?             yang:counter32
|  |  |  +--ro resv-tear-rcvd?             yang:counter32
|  |  |  +--ro summary-refresh-sent?       yang:counter32
|  |  |  +--ro summary-refresh-rcvd?       yang:counter32
|  |  |  +--ro unknown-recv?               yang:counter32
|  |  +--ro errors-stats
|  |     +--ro authenticate?   yang:counter64
|  |     +--ro checksum?       yang:counter64
|  |     +--ro packet-len?     yang:counter64
|  +--rw interface* [interface]
|     +--rw interface    if:interface-ref
|     +--rw config
|     |  +--rw refresh-reduction
|     |  |  +--rw enabled?   boolean
|     |  +--rw rsvp-hellos
|     |  |  +--rw enabled?   boolean
|     |  +--rw authentication
|     |     +--rw enabled?     boolean
|     |     +--rw password?    string
|     |     +--rw algorithm?   identityref
|     +--ro state
|        +--ro refresh-reduction
|        |  +--ro enabled?   boolean
|        +--ro rsvp-hellos
|        |  +--ro enabled?   boolean
|        +--ro authentication
|        |  +--ro enabled?     boolean
|        |  +--ro password?    string
|        |  +--ro algorithm?   identityref
|        +--ro packets-stats
|        |  +--ro sent?   yang:counter32
|        |  +--ro rcvd?   yang:counter32
|        +--ro protocol-stats
|        |  +--ro ack-sent?                  yang:counter32
|        |  +--ro ack-rcvd?                  yang:counter32
|        |  +--ro bundle-sent?               yang:counter32
|        |  +--ro bundle-rcvd?               yang:counter32
|        |  +--ro hello-sent?                yang:counter32
|        |  +--ro hello-rcvd?                yang:counter32
```

```
|        |  +--ro integrity-challenge-sent?   yang:counter32
|        |  +--ro integrity-challenge-rcvd?   yang:counter32
|        |  +--ro integrity-response-sent?    yang:counter32
|        |  +--ro integrity-response-rcvd?    yang:counter32
|        |  +--ro notify-sent?                yang:counter32
|        |  +--ro notify-rcvd?                yang:counter32
|        |  +--ro path-sent?                  yang:counter32
|        |  +--ro path-rcvd?                  yang:counter32
|        |  +--ro path-err-sent?              yang:counter32
|        |  +--ro path-err-rcvd?              yang:counter32
|        |  +--ro path-tear-sent?             yang:counter32
|        |  +--ro path-tear-rcvd?             yang:counter32
|        |  +--ro resv-sent?                  yang:counter32
|        |  +--ro resv-rcvd?                  yang:counter32
|        |  +--ro resv-confirm-sent?          yang:counter32
|        |  +--ro resv-confirm-rcvd?          yang:counter32
|        |  +--ro resv-err-sent?              yang:counter32
|        |  +--ro resv-err-rcvd?              yang:counter32
|        |  +--ro resv-tear-sent?             yang:counter32
|        |  +--ro resv-tear-rcvd?             yang:counter32
|        |  +--ro summary-refresh-sent?       yang:counter32
|        |  +--ro summary-refresh-rcvd?       yang:counter32
|        |  +--ro unknown-recv?               yang:counter32
|        +--ro errors-stats
|           +--ro authenticate?   yang:counter64
|           +--ro checksum?       yang:counter64
|           +--ro packet-len?     yang:counter64
```

3.1.1.3. Session Data

This branch of the data model covers configuration of elements relevant to RSVP neighbors. This would be discussed in detail in future revisions.

```
module: ietf-rsvp
   +--rw rsvp!
      +--rw sessions
      |  +--rw session* [dst_port source destination]
      |     +--rw dst_port       uint16
      |     +--rw source         inet:ip-address
      |     +--rw destination    inet:ip-address
      |     +--rw config
      |     +--ro state
```

3.1.1.4. Neighbor Data

This branch of the data model covers configuration of elements relevant to RSVP sessions. This would be discussed in detail in future revisions.

```
module: ietf-rsvp
   +--rw rsvp!
      +--rw neighbors
         +--rw neighbor* [address]
            +--rw address                inet:ip-address
            +--rw neighbor-attributes
               +--rw config
               +--ro state
                  +--ro epoch?                       uint32
                  +--ro expiry-time?                 uint32
                  +--ro graceful-restart
                  |  +--ro enabled?                boolean
                  |  +--ro local-restart-time?     uint32
                  |  +--ro local-recovery-time?    uint32
                  |  +--ro nbr-restart-time?       uint32
                  |  +--ro nbr-recovery-time?      uint32
                  |  +--ro helper-mode
                  |     +--ro helper-mode?                boolean
                  |     +--ro max-helper-restart-time?    uint32
                  |     +--ro max-helper-recovery-time?   uint32
                  |     +--ro nbr-restart-ttd?            uint32
                  |     +--ro nbr-recovery-ttd?           uint32
                  +--ro hello-status? enumeration {hellos}?
                  +--ro interface?                   if:interface-ref
                  +--ro neighbor-state?              enumeration
                  +--ro psb-count?                   uint32
                  +--ro rsb-count?                   uint32
                  +--ro refresh-reduction-capable?   boolean
                  +--ro restart-count?               uint32
                  +--ro restart-time?                yang:date-and-time
```

3.1.2. RPC and Notification Data

TBD.

3.1.3. YANG Module

<CODE BEGINS> file "ietf-rsvp@2015-10-16.yang"

```
module ietf-rsvp {

  namespace "urn:ietf:params:xml:ns:yang:ietf-rsvp";
```

```
    /* Replace with IANA when assigned */
    prefix "rsvp";

    import ietf-interfaces {
      prefix "if";
    }

    import ietf-inet-types {
      prefix inet;
    }

    import ietf-yang-types {
      prefix "yang";
    }

    organization
      "IETF TEAS Working Group";

    contact "TBA";

    description
      "This module contains the RSVP YANG data model.";

    revision 2015-10-16 {
      description "Latest revision of RSVP yang module.";
      reference "RFC2205";
    }

    identity hash-algorithm {
      description
        "Base identity for message-digest algorithm";
    }

    identity MD5 {
      base hash-algorithm;
      description
        "MD5 hash algorithm";
      reference "RFC1321";
    }

    identity SHA-1 {
      base hash-algorithm;
      description
        "SHA-1 hash algorithm";
      reference "NIST, FIPS PUB 180-1: Secure Hash Standard";
    }

    grouping graceful-restart_config {
```

```
      description
        "Base configuration parameters relating to RSVP
         Graceful-Restart";
      leaf enabled {
        type boolean;
        description
          "'true' if RSVP Graceful Restart is enabled.
           'false' if RSVP Graceful Restart is disabled.";
      }
    }

    grouping graceful-restart {
      description
        "Top level grouping for RSVP graceful-restart
        parameters";
      container graceful-restart {
        description
          "Top level container for RSVP graceful-restart";
        uses graceful-restart_config;
      }
    }

    grouping refresh-reduction_config {
      description
        "Configuration parameters relating to RSVP
        refresh reduction";

      leaf enabled {
        type boolean;
        description
          "'true' if RSVP Refresh Reduction is enabled.
           'false' if RSVP Refresh Reduction is disabled.";
      }
    }

    grouping refresh-reduction {
      description
        "Top level grouping for RSVP refresh reduction
        parameters";
      container refresh-reduction {
        description
          "Top level container for RSVP refresh reduction
          parameters";
        uses refresh-reduction_config;
      }
    }

    grouping authentication_config {
```

```
      description
        "Configuration parameters relating to RSVP
        authentication";
      leaf enabled {
        type boolean;
        description
          "'true' if RSVP Authenticaiton is enabled.
           'false' if RSVP Authenticaiton is disabled.";
      }
      leaf password {
        type string;
        description
          "An authentication key string";
      }
      leaf algorithm {
        type identityref {
          base hash-algorithm;
        }
        description
          "Cryptographic hash algorithm";
      }
    }

    grouping authentication {
      description
        "Top level grouping for RSVP authentication parameters";
      container authentication {
        description
          "Top level container for RSVP authentication
          parameters";
        uses authentication_config;
      }
    }

    grouping rsvp-hellos_config {
      description
        "Configuration parameters relating to RSVP
        hellos";
      leaf enabled {
        type boolean;
        description
          "'true' if RSVP Hello is enabled.
           'false' if RSVP Hello is disabled.";
      }
    }

    grouping rsvp-hellos {
      description
```

```
        "Top level grouping for RSVP hellos parameters";
      container rsvp-hellos {
        description
          "Top level container for RSVP hello parameters";
        uses rsvp-hellos_config;
      }
    }

    grouping signaling-parameters_config {
      description
        "Configuration parameters relating to RSVP
        signaling";
    }

    grouping signaling-parameters {
      description
        "Top level grouping for RSVP signaling  parameters";
      uses signaling-parameters_config;
    }

    grouping session-attributes {
      description
        "Top level grouping for RSVP session properties";
      container config {
        description
          "Configuration for session properties";
      }
      container state {
        config false;
        description
          "State information associated with RSVP
           session properties";
      }
    }

    grouping neighbor-attributes {
      description
        "Top level grouping for RSVP neighbor properties";
      container neighbor-attributes {
        description
          "Top level container for RSVP neighbor properties";
        container config {
          description
            "Configuration for neighbor properties";
        }
        container state {
          config false;
          description
```

```
            "State information associated with RSVP
             neighbor properties";
          uses neighbor-derived_state;
        }
      }
    }

    grouping statistics-packets_state {
      description
        "Packet statistics grouping";
      container packets-stats {
        description
          "Packet statistics container";
        leaf sent {
          type yang:counter32;
          description
            "Packet sent count";
        }

        leaf rcvd {
          type yang:counter32;
          description
            "Packet received count";
        }
      }
    }

    grouping statistics-protocol_state {
      description
        "RSVP protocol statistics grouping";
      container protocol-stats {
        description
          "RSVP protocol statistics container";
        leaf ack-sent {
          type yang:counter32;
          description
            "Hello sent count";
        }

        leaf ack-rcvd {
          type yang:counter32;
          description
            "Hello received count";
        }

        leaf bundle-sent {
          type yang:counter32;
          description
```

```
            "Bundle sent count";
          }

          leaf bundle-rcvd {
            type yang:counter32;
            description
              "Bundle received count";
          }

          leaf hello-sent {
            type yang:counter32;
            description
              "Hello sent count";
          }

          leaf hello-rcvd {
            type yang:counter32;
            description
              "Hello received count";
          }

          leaf integrity-challenge-sent {
            type yang:counter32;
            description
              "Integrity Challenge sent count";
          }

          leaf integrity-challenge-rcvd {
            type yang:counter32;
            description
              "Integrity Challenge received count";
          }

          leaf integrity-response-sent {
            type yang:counter32;
            description
              "Integrity Response sent count";
          }

          leaf integrity-response-rcvd {
            type yang:counter32;
            description
              "Integrity Response received count";
          }

          leaf notify-sent {
            type yang:counter32;
            description
```

```
            "Notify sent count";
          }

          leaf notify-rcvd {
            type yang:counter32;
            description
              "Notify received count";
          }

          leaf path-sent {
            type yang:counter32;
            description
              "Path sent count";
          }

          leaf path-rcvd {
            type yang:counter32;
            description
              "Path received count";
          }

          leaf path-err-sent {
            type yang:counter32;
            description
              "Path error sent count";
          }

          leaf path-err-rcvd {
            type yang:counter32;
            description
              "Path error received count";
          }

          leaf path-tear-sent {
            type yang:counter32;
            description
              "Path tear sent count";
          }

          leaf path-tear-rcvd {
            type yang:counter32;
            description
              "Path tear received count";
          }

          leaf resv-sent {
            type yang:counter32;
            description
```

```
            "Resv sent count";
          }

          leaf resv-rcvd {
            type yang:counter32;
            description
              "Resv received count";
          }

          leaf resv-confirm-sent {
            type yang:counter32;
            description
              "Confirm sent count";
          }

          leaf resv-confirm-rcvd {
            type yang:counter32;
            description
              "Confirm received count";
          }

          leaf resv-err-sent {
            type yang:counter32;
            description
              "Resv error sent count";
          }

          leaf resv-err-rcvd {
            type yang:counter32;
            description
              "Resv error received count";
          }

          leaf resv-tear-sent {
            type yang:counter32;
            description
              "Resv tear sent count";
          }

          leaf resv-tear-rcvd {
            type yang:counter32;
            description
              "Resv tear received count";
          }

          leaf summary-refresh-sent {
            type yang:counter32;
            description
```

```
            "Summary refresh sent count";
        }

        leaf summary-refresh-rcvd {
          type yang:counter32;
          description
            "Summary refresh received count";
        }

        leaf unknown-recv {
          type yang:counter32;
          description
            "Unknown packet received count";
        }
      }
    }

    grouping statistics-errors_state {
      description
        "Error statistics state grouping";
      container errors-stats {
        description
          "Error statistics state container";
        leaf authenticate {
          type yang:counter64;
          description
            "The total number of packets received with an
            authentication failure.";
        }

        leaf checksum {
          type yang:counter64;
          description
            "The total number of packets received with an invalid
            checksum value.";
        }

        leaf packet-len {
          type yang:counter64;
          description
            "The total number of packets received with an invalid
            packet length.";
        }
      }
    } // statistics_state

    grouping statistics_state {
      description "RSVP statistic attributes.";
```

```
      uses statistics-packets_state;
      uses statistics-protocol_state;
      uses statistics-errors_state;
    }

    grouping neighbor-derived_state {
      description
        "Derived state at neighbor level.";

      leaf epoch {
        type uint32;
        description
          "Neighbor epoch.";
      }

      leaf expiry-time {
        type uint32;
        units seconds;
        description
          "Neighbor expiry time after which the neighbor state
          is purged if no states associated with it";
      }

      container graceful-restart {
        description
          "Graceful restart information.";

        leaf enabled {
          type boolean;
          description
            "'true' if graceful restart is enabled for the
            neighbor.";
        }

        leaf local-restart-time {
          type uint32;
          units seconds;
          description
            "Local node restart time";
        }

        leaf local-recovery-time {
          type uint32;
          units seconds;
          description
            "Local node recover time";
        }
```

```
        leaf nbr-restart-time {
          type uint32;
          units seconds;
          description
            "Neighbor restart time";
        }

        leaf nbr-recovery-time {
          type uint32;
          units seconds;
          description
            "Neighbor recover time";
        }

        container helper-mode {
          description
            "Helper mode information ";

          leaf helper-mode {
            type boolean;
            description
              "'true' if helper mode is enabled.";
          }

          leaf max-helper-restart-time {
            type uint32;
            units seconds;
            description
              "The time the router or switch waits after it
               discovers that a neighboring router has gone down
               before it declares the neighbor down";
          }

          leaf max-helper-recovery-time {
            type uint32;
            units seconds;
            description
              "The amount of time the router retains the state of its
               RSVP neighbors while they undergo a graceful restart";
          }

          leaf nbr-restart-ttd {
            type uint32;
            units seconds;
            description
              "Number of seconds remaining for neighbor to send
               Hello message after restart.";
          }
```

```
          leaf nbr-recovery-ttd {
            type uint32;
            units seconds;
            description
              "Number of seconds remaining for neighbor to
              refresh.";
          }
        } // helper-mode
      } // graceful-restart

      leaf hello-status {
        type enumeration {
          enum "enabled" {
            description
              "Enabled";
          }
          enum "disabled" {
            description
              "Disabled";
          }
          enum "restarting" {
            description
              "Restarting";
          }
        }
        description
          "Hello status";
      }

      leaf interface {
        type if:interface-ref;
        description
          "Interface to this neighbor.";
      }

      leaf neighbor-state {
        type enumeration {
          enum "up" {
            description
              "up";
          }
          enum "down" {
            description
              "down";
          }
          enum "hello-disable" {
            description
              "hello-disable";
```

```
          }
          enum "restarting" {
            description
              "restarting";
          }
        }
        description
          "Neighbor state";
      }

      leaf psb-count {
        type uint32;
        description
          "Number of PSB state currently referencing the
          neighbor.";
      }

      leaf rsb-count {
        type uint32;
        description
          "Number of RSB state currently referencing the
          neighbor.";
      }

      leaf refresh-reduction-capable {
        type boolean;
        description
          "Whether neighbor is refresh reduction capable.";
      }

      leaf restart-count {
        type uint32;
        description
          "Number of times this neighbor restart";
      }

      leaf restart-time {
        type yang:date-and-time;
        description
          "Last restart time of the neighbor";
      }
    } // neighbor-derived_state

    grouping global-attributes {
      description
        "Top level grouping for RSVP global properties";
      container config {
        description
```

```
          "Configuration globals properties";
        uses graceful-restart;
      }
      container state {
        config false;
        description
          "State information associated with RSVP
           global properties";
        uses graceful-restart;
        uses statistics_state;
      }
    }

    grouping intf-attributes {
      description
        "Top level grouping for RSVP interface properties";
      container config {
        description
          "Configuration parameters relating to
          RSVP interface parameters";
        uses signaling-parameters;
        uses refresh-reduction;
        uses rsvp-hellos;
        uses authentication;
      }
      container state {
        config false;
        description
          "State information associated with RSVP
           interface parameters";
        uses signaling-parameters;
        uses refresh-reduction;
        uses rsvp-hellos;
        uses authentication;
        uses statistics_state;
      }
    }

    container rsvp {
      presence "Enable RSVP feature";
      description "RSVP feature container";
      container globals {
        description "RSVP global properties.";
        uses global-attributes;
      }

      container interfaces {
        description
```

```
        "RSVP interfaces container";
      uses intf-attributes;

      list interface {
        key "interface";
        description
          "RSVP interfaces.";
        leaf interface {
          type if:interface-ref;
          description
            "RSVP interface.";
        }
        uses intf-attributes;
      }
    }

    container sessions {
      description
        "RSVP sessions container";
      list session {
        key "dst_port source destination";
        description
          "List of RSVP sessions";
        leaf dst_port {
          type uint16;
          description "RSVP destination port";
          reference "RFC2205";
        }
        leaf source {
          type inet:ip-address;
          description "RSVP source address";
          reference "RFC2205";
        }
        leaf destination {
          type inet:ip-address;
          description "RSVP destination address";
          reference "RFC2205";
        }
        uses session-attributes;
      }
    }

    container neighbors {
      description
        "RSVP neighbors container";
      list neighbor {
        key "address";
        description
```

```
            "List of RSVP neighbors";
          leaf address {
            type inet:ip-address;
            description
              "Neighbor address";
          }
          uses neighbor-attributes;
        }
      }
    }
  }
```

<CODE ENDS>

3.2. RSVP Extended YANG Model

The RSVP extended YANG model covers optional configuration(s) for RSVP core feature(s). It also covers non-core RSVP feature(s) that are negotiated with "if-feature" checks.

3.2.1. Configuration and State Data

Figure 3 shows the YANG tree representation for configuration and state data that is augmenting the RSVP basic module:

```
module: ietf-rsvp-extended
augment /rsvp:rsvp/rsvp:globals/rsvp:config/
        rsvp:graceful-restart:
   +--rw restart-time?    uint32
   +--rw recovery-time?   uint32
augment /rsvp:rsvp/rsvp:globals/rsvp:state/
        rsvp:graceful-restart:
   +--ro restart-time?    uint32
   +--ro recovery-time?   uint32
augment /rsvp:rsvp/rsvp:globals/rsvp:state/rsvp:packets-stats:
   +--ro discontinuity-time?   yang:date-and-time
   +--ro tx-dropped?           yang:counter32
   +--ro rx-dropped?           yang:counter32
   +--ro tx-error?             yang:counter32
   +--ro rx-error?             yang:counter32
augment /rsvp:rsvp/rsvp:globals/rsvp:state/rsvp:protocol-stats:
augment /rsvp:rsvp/rsvp:globals/rsvp:state/rsvp:errors-stats:
augment /rsvp:rsvp/rsvp:interfaces/rsvp:config:
   +--rw refresh-interval?        uint32
   +--rw refresh-misses?          uint32
   +--rw checksum?                uint32
   +--rw patherr-state-removal?   empty
augment /rsvp:rsvp/rsvp:interfaces/rsvp:state:
```

```
      +--ro refresh-interval?        uint32
      +--ro refresh-misses?          uint32
      +--ro checksum?                uint32
      +--ro patherr-state-removal?   empty
   augment /rsvp:rsvp/rsvp:interfaces/rsvp:config/
         rsvp:refresh-reduction:
      +--rw bundle-message-max-size?    uint32
      +--rw reliable-ack-hold-time?     uint32
      +--rw reliable-ack-max-size?      uint32
      +--rw reliable-retransmit-time?   uint32
      +--rw reliable-srefresh?          empty
      +--rw summary-max-size?           uint32
   augment /rsvp:rsvp/rsvp:interfaces/rsvp:state/rsvp:refresh-reduction:
      +--ro bundle-message-max-size?    uint32
      +--ro reliable-ack-hold-time?     uint32
      +--ro reliable-ack-max-size?      uint32
      +--ro reliable-retransmit-time?   uint32
      +--ro reliable-srefresh?          empty
      +--ro summary-max-size?           uint32
   augment /rsvp:rsvp/rsvp:interfaces/rsvp:config/rsvp:rsvp-hellos:
      +--rw interface-based?   empty
      +--rw hello-interval?    uint32
      +--rw hello-misses?      uint32
   augment /rsvp:rsvp/rsvp:interfaces/rsvp:state/rsvp:rsvp-hellos:
      +--ro interface-based?   empty
      +--ro hello-interval?    uint32
      +--ro hello-misses?      uint32
   augment /rsvp:rsvp/rsvp:interfaces/rsvp:config/rsvp:authentication:
      +--rw lifetime?      uint32
      +--rw window-size?   uint32
      +--rw challenge?     empty
      +--rw retransmits?   uint32
      +--rw key-chain?     string
   augment /rsvp:rsvp/rsvp:interfaces/rsvp:state/rsvp:authentication:
      +--ro lifetime?      uint32
      +--ro window-size?   uint32
      +--ro challenge?     empty
      +--ro retransmits?   uint32
      +--ro key-chain?     string
   augment /rsvp:rsvp/rsvp:interfaces/rsvp:interface/rsvp:config:
      +--rw refresh-interval?        uint32
      +--rw refresh-misses?          uint32
      +--rw checksum?                uint32
      +--rw patherr-state-removal?   empty
   augment /rsvp:rsvp/rsvp:interfaces/rsvp:interface/rsvp:state:
      +--ro refresh-interval?        uint32
      +--ro refresh-misses?          uint32
      +--ro checksum?                uint32
```

```
      +--ro patherr-state-removal?   empty
   augment /rsvp:rsvp/rsvp:interfaces/rsvp:interface/rsvp:config/
           rsvp:refresh-reduction:
      +--rw bundle-message-max-size?    uint32
      +--rw reliable-ack-hold-time?     uint32
      +--rw reliable-ack-max-size?      uint32
      +--rw reliable-retransmit-time?   uint32
      +--rw reliable-srefresh?          empty
      +--rw summary-max-size?           uint32
   augment /rsvp:rsvp/rsvp:interfaces/rsvp:interface/rsvp:state/
           rsvp:refresh-reduction:
      +--ro bundle-message-max-size?    uint32
      +--ro reliable-ack-hold-time?     uint32
      +--ro reliable-ack-max-size?      uint32
      +--ro reliable-retransmit-time?   uint32
      +--ro reliable-srefresh?          empty
      +--ro summary-max-size?           uint32
   augment /rsvp:rsvp/rsvp:interfaces/rsvp:interface/rsvp:config/
           rsvp:rsvp-hellos:
      +--rw interface-based?   empty
      +--rw hello-interval?    uint32
      +--rw hello-misses?      uint32
   augment /rsvp:rsvp/rsvp:interfaces/rsvp:interface/rsvp:state/
           rsvp:rsvp-hellos:
      +--ro interface-based?   empty
      +--ro hello-interval?    uint32
      +--ro hello-misses?      uint32
   augment /rsvp:rsvp/rsvp:interfaces/rsvp:interface/rsvp:config/
           rsvp:authentication:
      +--rw lifetime?      uint32
      +--rw window-size?   uint32
      +--rw challenge?     empty
      +--rw retransmits?   uint32
      +--rw key-chain?     string
   augment /rsvp:rsvp/rsvp:interfaces/rsvp:interface/rsvp:state/
           rsvp:authentication:
      +--ro lifetime?      uint32
      +--ro window-size?   uint32
      +--ro challenge?     empty
      +--ro retransmits?   uint32
      +--ro key-chain?     string
```

Figure 3: RSVP extended YANG Tree representation

3.2.2. YANG Module

Figure 4 shows the RSVP extended YANG module:

<CODE BEGINS> file "ietf-rsvp-extended@2015-10-16.yang"

```
module ietf-rsvp-extended {

  namespace "urn:ietf:params:xml:ns:yang:ietf-rsvp-extended";

  prefix "rsvp-ext";

  import ietf-rsvp {
    prefix "rsvp";
  }

  import ietf-yang-types {
    prefix "yang";
  }

  organization
    "IETF TEAS Working Group";

  contact "TBA";

  description
    "This module contains the RSVP YANG data model.";

  revision 2015-10-16 {
    description "Latest revision of RSVP extensions yang module.";
    reference "RFC2205";
  }

  /* RSVP features */
  feature authentication {
    description
      "Indicates support for RSVP authentication";
  }

  feature error-statistics {
    description
      "Indicates support for error statistics";
  }

  feature global-statistics {
    description
      "Indicates support for global statistics";
  }
```

```
    feature graceful-restart {
      description
        "Indicates support for RSVP graceful restart";
    }

    feature hellos {
      description
        "Indicates support for RSVP hellos (RFC3209).";
    }

    feature notify {
      description
        "Indicates support for RSVP notify message (RFC3473).";
    }

    feature refresh-reduction {
      description
        "Indicates support for RSVP refresh reduction
        (RFC2961).";
    }

    feature refresh-reduction-extended {
      description
        "Indicates support for RSVP refresh reduction
        (RFC2961).";
    }

    feature per-interface-statistics {
      description
        "Indicates support for per interface statistics";
    }

    grouping graceful-restart-extended_config {
      description
        "Configuration parameters relating to RSVP
        Graceful-Restart";
      leaf restart-time {
        type uint32;
        units seconds;
        description
          "Graceful restart time (seconds).";
        reference
          "RFC 5495: Description of the Resource
          Reservation Protocol - Traffic-Engineered
          (RSVP-TE) Graceful Restart Procedures";
      }
      leaf recovery-time {
        type uint32;
```

```
        description
          "RSVP state recovery time";
      }
    }

    grouping authentication-extended_config {
      description
        "Configuration parameters relating to RSVP
        authentication";
      leaf lifetime {
        type uint32 {
          range "30..86400";
        }
        description
          "Life time for each security association";
        reference
          "RFC 2747: RSVP Cryptographic
          Authentication";
      }
      leaf window-size {
        type uint32 {
          range "1..64";
        }
        description
          "Window-size to limit number of out-of-order
          messages.";
        reference
          "RFC 2747: RSVP Cryptographic
          Authentication";
      }
      leaf challenge {
        type empty;
        description
          "Enable challenge messages.";
        reference
          "RFC 2747: RSVP Cryptographic
          Authentication";
      }
      leaf retransmits {
        type uint32 {
          range "1..10000";
        }
        description
          "Number of retransmits when messages are
          dropped.";
        reference
          "RFC 2747: RSVP Cryptographic
          Authentication";
```

```
      }
      leaf key-chain {
        type string {
          length "1..32";
        }
        description
          "Key chain name to authenticate RSVP
          signaling messages.";
        reference
          "RFC 2747: RSVP Cryptographic
          Authentication";
      }
    }

    grouping rsvp-hellos-extended_config {
      description
        "Configuration parameters relating to RSVP
        hellos";
      leaf interface-based {
        type empty;
        description "Enable interface-based
          Hello adjacency if present.";
      }
      leaf hello-interval {
        type uint32;
        units milliseconds;
        description
          "Configure interval between successive Hello
          messages in milliseconds.";
        reference
          "RFC 3209: RSVP-TE: Extensions to RSVP for
          LSP Tunnels.
          RFC 5495: Description of the Resource
          Reservation Protocol - Traffic-Engineered
          (RSVP-TE) Graceful Restart Procedures";
      }
      leaf hello-misses {
        type uint32 {
          range "1..10";
        }
        description
          "Configure max number of consecutive missed
          Hello messages.";
        reference
          "RFC 3209: RSVP-TE: Extensions to RSVP for
          LSP Tunnels RFC 5495: Description of the
          Resource Reservation Protocol - Traffic-
          Engineered (RSVP-TE) Graceful Restart
```

```
          Procedures";
      }
    }

    grouping signaling-parameters-extended_config {
      description
        "Configuration parameters relating to RSVP
        signaling";
      leaf refresh-interval {
        type uint32;
        description
          "Set interval between successive refreshes";
      }
      leaf refresh-misses {
        type uint32;
        description
          "Set max number of consecutive missed
          messages for state expiry";
      }
      leaf checksum {
        type uint32;
        description
          "Enable RSVP message checksum computation";
      }
      leaf patherr-state-removal {
        type empty;
        description
          "State-Removal flag in Path Error message
          if present.";
      }
    }

    grouping refresh-reduction-extended_config {
      description
        "Configuration parameters relating to RSVP
        refresh reduction";

      leaf bundle-message-max-size {
        type uint32 {
          range "512..65000";
        }
        description
          "Configure maximum size (bytes) of a
          single RSVP Bundle message.";
      }
      leaf reliable-ack-hold-time {
        type uint32;
        units milliseconds;
```

```
        description
          "Configure hold time in milliseconds for
          sending RSVP ACK message(s).";
      }
      leaf reliable-ack-max-size {
        type uint32;
        description
          "Configure max size of a single RSVP ACK
          message.";
      }
      leaf reliable-retransmit-time {
        type uint32;
        units milliseconds;
        description
          "Configure min delay in milliseconds to
          wait for an ACK before a retransmit.";
      }
      leaf reliable-srefresh {
        type empty;
        description
          "Configure use of reliable messaging for
          summary refresh if present.";
      }
      leaf summary-max-size {
        type uint32 {
          range "20..65000";
        }
        description
          "Configure max size (bytes) of a single
          RSVP summary refresh message.";
      }
    }

    grouping statistics-packets-extended_state {
      description
        "Packet statistics.";
      leaf discontinuity-time {
        type yang:date-and-time;
        description
          "The time on the most recent occasion at which any one
          or more of the statistic counters suffered a
          discontinuity. If no such discontinuities have occurred
          since the last re-initialization of the local
          management subsystem, then this node contains the time
          the local management subsystem re-initialized itself.";
      }
      leaf tx-dropped {
        type yang:counter32;
```

```
        description
          "Packet tx dropped count";
      }

      leaf rx-dropped {
        type yang:counter32;
        description
          "Packet rx dropped count";
      }

      leaf tx-error {
        type yang:counter32;
        description
          "Packet tx error count";
      }

      leaf rx-error {
        type yang:counter32;
        description
          "Packet rx error count";
      }
    }

    grouping statistics-protocol-extended_state {
      description
        "RSVP protocol statistics.";
    }

    grouping statistics-errors-extended_state {
      description
        "Error statistics.";
    }

    grouping statistics-extended_state {
      description "RSVP statistic attributes.";
      uses statistics-packets-extended_state;
      uses statistics-protocol-extended_state;
      uses statistics-errors-extended_state;
    }

    /**
     * RSVP extensions augmentations
     */

    /* RSVP globals graceful restart*/
    augment "/rsvp:rsvp/rsvp:globals/rsvp:config/"+
            "rsvp:graceful-restart" {
      description
```

```
        "RSVP globals configuration extensions";
      uses graceful-restart-extended_config;
    }
    augment "/rsvp:rsvp/rsvp:globals/rsvp:state/"+
            "rsvp:graceful-restart" {
      description
        "RSVP globals state extensions";
      uses graceful-restart-extended_config;
    }

    /* RSVP statistics augmentation */
    augment "/rsvp:rsvp/rsvp:globals/rsvp:state/"+
            "rsvp:packets-stats" {
      description
        "RSVP packet stats extensions";
      uses statistics-packets-extended_state;
    }
    augment "/rsvp:rsvp/rsvp:globals/rsvp:state/"+
            "rsvp:protocol-stats" {
      description
        "RSVP protocol message stats extensions";
      uses statistics-protocol-extended_state;
    }
    augment "/rsvp:rsvp/rsvp:globals/rsvp:state/"+
            "rsvp:errors-stats" {
      description
        "RSVP errors stats extensions";
      uses statistics-errors-extended_state;
    }

    /**
     * RSVP all interfaces extensions
     */

    /* RSVP interface signaling extensions */
    augment "/rsvp:rsvp/rsvp:interfaces/rsvp:config" {
      description
        "RSVP signaling all interfaces configuration extensions";
      uses signaling-parameters-extended_config;
    }
    augment "/rsvp:rsvp/rsvp:interfaces/rsvp:state" {
      description
        "RSVP signaling all interfaces state extensions";
      uses signaling-parameters-extended_config;
    }
    /* RSVP refresh reduction extension */
    augment "/rsvp:rsvp/rsvp:interfaces/rsvp:config/" +
      "rsvp:refresh-reduction" {
```

```
      description
        "RSVP refresh-reduction all interface configuration
         extensions";
      uses refresh-reduction-extended_config;
    }
    augment "/rsvp:rsvp/rsvp:interfaces/rsvp:state/" +
      "rsvp:refresh-reduction" {
      description
        "RSVP refresh-reduction all interfaces state extensions";
      uses refresh-reduction-extended_config;
    }

    /* RSVP hellos extension */
    augment "/rsvp:rsvp/rsvp:interfaces/rsvp:config/rsvp:rsvp-hellos" {
      description
        "RSVP hello all interfaces configuration extensions";
      uses rsvp-hellos-extended_config;
    }
    augment "/rsvp:rsvp/rsvp:interfaces/rsvp:state/rsvp:rsvp-hellos" {
      description
        "RSVP hello all interfaces state extensions";
      uses rsvp-hellos-extended_config;
    }

    /* RSVP authentication extension */
    augment "/rsvp:rsvp/rsvp:interfaces/rsvp:config/" +
      "rsvp:authentication" {
      description
        "RSVP authentication all interfaces configuration extensions";
      uses authentication-extended_config;
    }
    augment "/rsvp:rsvp/rsvp:interfaces/rsvp:state/" +
      "rsvp:authentication" {
      description
        "RSVP authentication all interfaces state extensions";
      uses authentication-extended_config;
    }

    /**
     * RSVP interface extensions
     */

    /* RSVP interface signaling extensions */
    augment "/rsvp:rsvp/rsvp:interfaces/rsvp:interface/rsvp:config" {
      description
        "RSVP signaling interface configuration extensions";
      uses signaling-parameters-extended_config;
    }
```

```
    augment "/rsvp:rsvp/rsvp:interfaces/rsvp:interface/rsvp:state" {
      description
        "RSVP signaling interface state extensions";
      uses signaling-parameters-extended_config;
    }

    /* RSVP refresh reduction extension */
    augment "/rsvp:rsvp/rsvp:interfaces/rsvp:interface/rsvp:config/" +
      "rsvp:refresh-reduction" {
      description
        "RSVP refresh-reduction interface configuration extensions";
      uses refresh-reduction-extended_config;
    }
    augment "/rsvp:rsvp/rsvp:interfaces/rsvp:interface/rsvp:state/" +
      "rsvp:refresh-reduction" {
      description
        "RSVP refresh-reduction interface state extensions";
      uses refresh-reduction-extended_config;
    }

    /* RSVP hellos extension */
    augment "/rsvp:rsvp/rsvp:interfaces/rsvp:interface/rsvp:config/" +
      "rsvp:rsvp-hellos" {
      description
        "RSVP hello interface configuration extensions";
      uses rsvp-hellos-extended_config;
    }
    augment "/rsvp:rsvp/rsvp:interfaces/rsvp:interface/rsvp:state/" +
      "rsvp:rsvp-hellos" {
      description
        "RSVP hello interface state extensions";
      uses rsvp-hellos-extended_config;
    }

    /* RSVP authentication extension */
    augment "/rsvp:rsvp/rsvp:interfaces/rsvp:interface/rsvp:config/" +
      "rsvp:authentication" {
      description
        "RSVP authentication interface configuration extensions";
      uses authentication-extended_config;
    }
    augment "/rsvp:rsvp/rsvp:interfaces/rsvp:interface/rsvp:state/" +
      "rsvp:authentication" {
      description
        "RSVP authentication interface state extensions";
      uses authentication-extended_config;
    }
  }
```

<CODE ENDS>

Figure 4: RSVP extended YANG module

3.3. RSVP-TE Generic YANG Model

This section contains the augmentation of the RSVP base YANG model for signalling Traffic-Engineering (RSVP-TE) Label Switched Paths (LSPs). New module is introduced that augment the RSVP-TE generic module to cover data items that are technology agnostic.

This model imports and augments the base RSVP YANG model (presented in Section 3.1.3). It also imports and augments the TE YANG model defined in [I-D.ietf-teas-yang-te] to enable configuration of RSVP-TE attributes on TE tunnels.

The following subsections provide overview of the parts of the RSVP-TE generic model pertaining to configuration and state data.

3.3.1. Configuration and State Data

There are three types of configuration and state data nodes in this module:

- those augmenting or extending the base RSVP module
- those augmenting or extending the base TE generic module
- those that are specific to the RSVP-TE module

Below is a YANG tree representation for data items defined in the RSVP-TE generic module:

```
module: ietf-rsvp-te
augment /rsvp:rsvp/rsvp:globals:
augment /rsvp:rsvp/rsvp:interfaces:
augment /rsvp:rsvp/rsvp:interfaces/rsvp:interface:
augment /rsvp:rsvp/rsvp:sessions:
augment /rsvp:rsvp/rsvp:neighbors:
augment /ietf-te:te/ietf-te:tunnels/ietf-te:tunnel:
   +--rw config
   |  +--rw lsp-source?                 inet:ip-address
   |  +--rw lsp-signaled-name?          string
   |  +--rw lsp-priority-setup?         uint8
   |  +--rw lsp-priority-hold?          uint8
   |  +--rw local-recording-desired?    empty
   |  +--rw se-style-desired?           empty
   |  +--rw path-reevaluation-request?  empty
```

```
      |  +--rw soft-preemption-desired?     empty
      |  +--rw lsp-rerouting?               enumeration
      |  +--rw lsp-integrety-required?      empty
      |  +--rw lsp-contiguous?              empty
      |  +--rw lsp-stitching-desired?       empty
      |  +--rw lsp-preplanned?              empty
      |  +--rw lsp-oob-mapping?             empty
      +--ro state
         +--ro lsp-source?                  inet:ip-address
         +--ro lsp-signaled-name?           string
         +--ro lsp-priority-setup?          uint8
         +--ro lsp-priority-hold?           uint8
         +--ro local-recording-desired?     empty
         +--ro se-style-desired?            empty
         +--ro path-reevaluation-request?   empty
         +--ro soft-preemption-desired?     empty
         +--ro lsp-rerouting?               enumeration
         +--ro lsp-integrety-required?      empty
         +--ro lsp-contiguous?              empty
         +--ro lsp-stitching-desired?       empty
         +--ro lsp-preplanned?              empty
         +--ro lsp-oob-mapping?             empty
   augment /ietf-te:te/ietf-te:interfaces/ietf-te:interface:
   augment /ietf-te:te/ietf-te:lsps-state/ietf-te:lsp:
      +--ro lsp-source?                  inet:ip-address
      +--ro lsp-signaled-name?           string
      +--ro lsp-priority-setup?          uint8
      +--ro lsp-priority-hold?           uint8
      +--ro local-recording-desired?     empty
      +--ro se-style-desired?            empty
      +--ro path-reevaluation-request?   empty
      +--ro soft-preemption-desired?     empty
      +--ro lsp-rerouting?               enumeration
      +--ro lsp-integrety-required?      empty
      +--ro lsp-contiguous?              empty
      +--ro lsp-stitching-desired?       empty
      +--ro lsp-preplanned?              empty
      +--ro lsp-oob-mapping?             empty
      +--ro explicit-route-object
      |  +--ro incoming-explicit-route-subobjects* [subobject-index]
      |  |  +--ro subobject-index     uint32
      |  |  +--ro (type)?
      |  |     +--:(ipv4-address)
      |  |     |  +--ro v4-address?         inet:ipv4-address
      |  |     |  +--ro v4-prefix-length?   uint8
      |  |     |  +--ro v4-loose?           boolean
      |  |     +--:(ipv6-address)
      |  |     |  +--ro v6-address?         inet:ipv6-address
```

```
      |  |     |  +--ro v6-prefix-length?   uint8
      |  |     |  +--ro v6-loose?           boolean
      |  |     +--:(as-number)
      |  |     |  +--ro as-number?          uint16
      |  |     +--:(unnumbered-link)
      |  |     |  +--ro router-id?          inet:ip-address
      |  |     |  +--ro interface-id?       uint32
      |  |     +--:(label)
      |  |        +--ro value?              uint32
      |  +--ro outgoing-explicit-route-subobjects* [subobject-index]
      |     +--ro subobject-index     uint32
      |     +--ro (type)?
      |        +--:(ipv4-address)
      |        |  +--ro v4-address?         inet:ipv4-address
      |        |  +--ro v4-prefix-length?   uint8
      |        |  +--ro v4-loose?           boolean
      |        +--:(ipv6-address)
      |        |  +--ro v6-address?         inet:ipv6-address
      |        |  +--ro v6-prefix-length?   uint8
      |        |  +--ro v6-loose?           boolean
      |        +--:(as-number)
      |        |  +--ro as-number?          uint16
      |        +--:(unnumbered-link)
      |        |  +--ro router-id?          inet:ip-address
      |        |  +--ro interface-id?       uint32
      |        +--:(label)
      |           +--ro value?              uint32
      +--ro record-route-object
         +--ro path-record-route-subobjects* [subobject-index]
         |  +--ro subobject-index     uint32
         |  +--ro (type)?
         |     +--:(ipv4-address)
         |     |  +--ro v4-address?         inet:ipv4-address
         |     |  +--ro v4-prefix-length?   uint8
         |     |  +--ro v4-flags?           uint8
         |     +--:(ipv6-address)
         |     |  +--ro v6-address?         inet:ipv6-address
         |     |  +--ro v6-prefix-length?   uint8
         |     |  +--ro v6-flags?           uint8
         |     +--:(label)
         |        +--ro value?              uint32
         |        +--ro flags?              uint8
         +--ro resv-record-route-subobjects* [subobject-index]
            +--ro subobject-index     uint32
            +--ro (type)?
               +--:(ipv4-address)
               |  +--ro v4-address?         inet:ipv4-address
               |  +--ro v4-prefix-length?   uint8
```

```
               |  +--ro v4-flags?           uint8
               +--:(ipv6-address)
               |  +--ro v6-address?         inet:ipv6-address
               |  +--ro v6-prefix-length?   uint8
               |  +--ro v6-flags?           uint8
               +--:(label)
                  +--ro value?             uint32
                  +--ro flags?             uint8
```

Figure 5: RSVP-TE YANG Tree representation

3.3.2. RPC and Notification Data

TBD.

3.3.3. YANG Module

<CODE BEGINS> file "ietf-rsvp-te@2015-10-16.yang"

```
module ietf-rsvp-te {

  namespace "urn:ietf:params:xml:ns:yang:ietf-rsvp-te";

  prefix "rsvp-te";

  import ietf-rsvp {
    prefix rsvp;
  }

  import ietf-te {
    prefix ietf-te;
  }

  import ietf-inet-types {
    prefix inet;
  }

  /* Import TE generic types */
  import ietf-te-types {
    prefix ietf-te-types;
  }

  organization
    "IETF TEAS Working Group";

  contact "TBA";

  description
```

```
       "This module contains the RSVP YANG generic data model.";

     revision 2015-10-16 {
       description "Latest revision to RSVP-TE generic YANG module";
       reference "RFC3209";
     }

     /**
      * RSVP-TE LSPs groupings.
      */

     grouping lsp-record-route-information_state {
       description "recorded route information grouping";
       container record-route-object {
         description "RSVP recorded route object information";
         list path-record-route-subobjects {
           when "../origin-type != 'ingress'" {
             description "Applicable on non-ingress LSPs only";
           }
           key "subobject-index";
           description "";
           leaf subobject-index {
             type uint32;
             description "RRO subobject index";
           }
           uses ietf-te-types:record-route-subobject;
         }
         list resv-record-route-subobjects {
           when "../origin-type != 'egress'" {
             description "Applicable on non-egress LSPs only";
           }
           key "subobject-index";
           description "";
           leaf subobject-index {
             type uint32;
             description "RRO subobject index";
           }
           uses ietf-te-types:record-route-subobject;
         }
       }
     }

     grouping lsp-explicit-route-information_state {
       description "RSVP-TE LSP explicit-route information";
       container explicit-route-object {
         description "Explicit route object information";
         list incoming-explicit-route-subobjects {
           when "../origin-type != 'ingress'" {
```

```
            description "Applicable on non-ingress LSPs only";
          }
          key "subobject-index";
          description "";
          leaf subobject-index {
            type uint32;
            description "ERO subobject index";
          }
          uses ietf-te-types:explicit-route-subobject;
        }
        list outgoing-explicit-route-subobjects {
          when "../origin-type != 'egress'" {
            description "Applicable on non-egress LSPs only";
          }
          key "subobject-index";
          description "";
          leaf subobject-index {
            type uint32;
            description "ERO subobject index";
          }
          uses ietf-te-types:explicit-route-subobject;
        }
      }
    }

    grouping lsp-attributes-flags_config {
      description
        "Configuration parameters relating to RSVP-TE LSP
        attribute flags";
      leaf lsp-rerouting {
        type enumeration {
          enum end-to-end-routing {
            description
              "End-to-end routing desired";
            reference "RFC4920, RFC5420";
          }
          enum boundary-rerouting {
            description
              "Boundary rerouting desired";
            reference "RFC4920, RFC5420";
          }
          enum segment-based-rerouting {
            description
              "Segment-based rerouting desired";
            reference "RFC4920, RFC5420";
          }
        }
        description "LSP rerouting types";
```

```
      }
      leaf lsp-integrety-required {
        type empty;
        description "LSP integrity desired";
        reference "RFC4875";
      }
      leaf lsp-contiguous {
        type empty;
        description "Contiguous LSP";
        reference "RFC5151";
      }
      leaf lsp-stitching-desired {
        type empty;
        description "Stitched LSP";
        reference "RFC5150";
      }
      leaf lsp-preplanned {
        type empty;
        description "Preplanned LSP";
        reference "RFC6001";
      }
      leaf lsp-oob-mapping {
        type empty;
        description
          "Mapping is done out-of-band";
        reference "RFC6511";
      }
    }

    grouping lsp-session-attributes-obj-flags_config {
      description
        "Configuration parameters relating to RSVP-TE LSP
        session attribute flags";
      reference
        "RFC4859: Registry for RSVP-TE Session Flags";
      leaf local-recording-desired {
        type empty;
        description "Path recording is desired.";
        reference "RFC3209";
      }
      leaf se-style-desired {
        type empty;
        description "SE Style desired";
        reference "RFC3209";
      }
      leaf path-reevaluation-request {
        type empty;
        description "Path re-evaluation request";
```

```
        reference "RFC4736";
      }
      leaf soft-preemption-desired {
        type empty;
        description "Soft-preemption is desired";
        reference "RFC5712";
      }
    }

    grouping lsp-properties_config {
      description
        "Configuration parameters relating to RSVP-TE LSP
        session attribute flags";
      leaf lsp-source {
        type inet:ip-address;
        description
          "LSP source address.";
      }
      leaf lsp-signaled-name {
        type string;
        description
          "Sets the session name to use in the session
          attribute object.";
      }
      leaf lsp-priority-setup {
        type uint8 {
          range "0..7";
        }
        description
          "RSVP session attributes setup priority";
      }
      leaf lsp-priority-hold {
        type uint8 {
          range "0..7";
        }
        description
          "RSVP session attributes hold priority";
      }
      uses lsp-session-attributes-obj-flags_config;
      uses lsp-attributes-flags_config;
    }

    grouping tunnel-properties {
      description
        "Top level grouping for LSP properties.";
      container config {
        description
          "Configuration parameters relating to
```

```
          LSP properties";
        uses lsp-properties_config;
      }
      container state {
        config false;
        description
          "State information associated with LSP
          properties";
        uses lsp-properties_config;
      }
    }
    /*** End of RSVP-TE LSP groupings ***/

    /**
     * RSVP-TE generic global properties.
     */

    grouping global-soft-preemption_config {
      description
        "Configuration for global RSVP-TE soft preemption";
      leaf soft-preemption-timeout {
        type uint16 {
          range 0..300;
        }
        default 0;
        description
          "Timeout value for soft preemption to revert
          to hard preemption";
      }
    }

    grouping global-soft-preemption {
      description
        "Top level group for RSVP-TE soft-preemption";
      container global-soft-preemption {
        presence "Enables soft preemption on a node.";
        description
          "Top level container for RSVP-TE soft-preemption";
        container config {
          description
            "Configuration parameters relating to RSVP
            soft preemption support";
          uses global-soft-preemption_config;
        }
        container state {
          description "State parameters relating to RSVP
            soft preemption support";
          uses global-soft-preemption_config;
```

```
        }
      }
    }
    /*** End of RSVP-TE generic global properties. ***/

    /**
     * RSVP-TE interface generic groupings.
     */

    grouping rsvp-te-interface-attributes {
      description
        "Top level grouping for RSVP-TE interface properties.";
      container rsvp-te-interface-attributes {
        description
          "Top level container for RSVP-TE interface
          properties";
        container config {
          description
            "Configuration parameters relating to RSVP-TE
            bandwidth";
        }
        container state {
          config false;
          description
            "State information associated with RSVP-TE
            bandwidth";
        }
      }
    }
    /*** End of RSVP-TE generic groupings ***/


    /* RSVP-TE global propeerties */
    augment "/rsvp:rsvp/rsvp:globals" {
      description
        "RSVP-TE augmentation to RSVP globals";
    }

    /* Linkage to the base RSVP all links */
    augment "/rsvp:rsvp/rsvp:interfaces" {
      description
        "RSVP-TE generic data augmentation pertaining to interfaces";
      /* To be added */
    }

    /* Linkage to per RSVP interface */
    augment "/rsvp:rsvp/rsvp:interfaces/rsvp:interface" {
      description
```

```
        "RSVP-TE generic data augmentation pertaining to specific
         interface";
      /* To be added */
    }

    /* add augmentation for sessions neighbors */
    augment "/rsvp:rsvp/rsvp:sessions" {
      description
        "RSVP-TE generic data augmentation pertaining to session";
      /* To be added */
    }

    augment "/rsvp:rsvp/rsvp:neighbors" {
      description
        "RSVP-TE generic data augmentation pertaining to neighbors";
      /* To be added */
    }

    /**
     * RSVP-TE generic augmentations of generic TE model.
     */

    augment "/ietf-te:te/ietf-te:tunnels/ietf-te:tunnel" {
      description
        "RSVP-TE generic data augmentation pertaining to TE tunnels";
      uses tunnel-properties;
    }

    augment "/ietf-te:te/ietf-te:interfaces/ietf-te:interface" {
      description
        "RSVP-TE generic data augmentation pertaining to specific TE
         interface";
    }

    augment "/ietf-te:te/ietf-te:lsps-state/ietf-te:lsp" {
      description
        "RSVP-TE generic data augmentation pertaining to specific TE
         LSP";
      uses lsp-properties_config;
      uses lsp-explicit-route-information_state;
      uses lsp-record-route-information_state;
    }
  }

  <CODE ENDS>
```

3.4. RSVP-TE MPLS Packet Model

This section describes the MPLS packet RSVP-TE YANG module that augments the RSVP-TE generic module to signal packet MPLS LSPs. RSVP-TE YANG modules for other dataplane technologies (e.g. OTN or WDM) will be defined in separate modules and in other drafts.

The following subsections describe the configuration and state data . pertaining to RSVP-TE packet MPLS YANG data model.

3.4.1. Configuration and State Data

The following are possible types of configuration and state data nodes in this module:

- those augmenting or extending the generic RSVP-TE module
- those augmenting or extending the base TE generic module
- those that are specific to the RSVP-TE packet module

Below is a YANG tree representation for data items defined in the RSVP-TE packet MPLS module:

```
module: ietf-rsvp-te-psc
augment /rsvp:rsvp/rsvp:globals:
   +--rw config
   |  +--rw rsvp-frr-local-revert!
   |     +--rw rsvp-frr-local-revert-delay?   uint32
   +--ro state
      +--ro rsvp-frr-local-revert!
         +--ro rsvp-frr-local-revert-delay?   uint32
augment /rsvp:rsvp/rsvp:interfaces:
augment /rsvp:rsvp/rsvp:interfaces/rsvp:interface:
augment /rsvp:rsvp/rsvp:sessions:
augment /rsvp:rsvp/rsvp:neighbors:
augment /ietf-te:te/ietf-te:tunnels/ietf-te:tunnel:
   +--rw config
   |  +--rw static-bandwidth?               uint32
   |  +--rw auto-bandwidth
   |  |  +--rw enabled?            boolean
   |  |  +--rw min-bw?             uint32
   |  |  +--rw max-bw?             uint32
   |  |  +--rw adjust-interval?    uint32
   |  |  +--rw adjust-threshold?   uint32
   |  |  +--rw overflow
   |  |  |  +--rw enabled?               boolean
   |  |  |  +--rw overflow-threshold?    uint32
```

```
   |  |  |  +--rw trigger-event-count?   uint16
   |  |  +--rw underflow
   |  |     +--rw enabled?               boolean
   |  |     +--rw underflow-threshold?   uint32
   |  |     +--rw trigger-event-count?   uint16
   |  +--rw local-protection-desired?       empty
   |  +--rw bandwidth-protection-desired?   empty
   |  +--rw node-protection-desired?        empty
   |  +--rw non-php-desired?                empty
   |  +--rw entropy-label-cap?              empty
   |  +--rw oam-mep-entities-desired?       empty
   |  +--rw oam-mip-entities-desired?       empty
   +--ro state
      +--ro static-bandwidth?              uint32
      +--ro auto-bandwidth
      |  +--ro enabled?            boolean
      |  +--ro min-bw?             uint32
      |  +--ro max-bw?             uint32
      |  +--ro adjust-interval?    uint32
      |  +--ro adjust-threshold?   uint32
      |  +--ro overflow
      |  |  +--ro enabled?               boolean
      |  |  +--ro overflow-threshold?    uint32
      |  |  +--ro trigger-event-count?   uint16
      |  +--ro underflow
      |     +--ro enabled?               boolean
      |     +--ro underflow-threshold?   uint32
      |     +--ro trigger-event-count?   uint16
      +--ro local-protection-desired?       empty
      +--ro bandwidth-protection-desired?   empty
      +--ro node-protection-desired?        empty
      +--ro non-php-desired?                empty
      +--ro entropy-label-cap?              empty
      +--ro oam-mep-entities-desired?       empty
      +--ro oam-mip-entities-desired?       empty
augment /ietf-te:te/ietf-te:interfaces/ietf-te:interface:
   +--rw config
   |  +--rw (bandwidth-value)?
   |  |  +--:(absolute)
   |  |  |  +--rw absolute-value?   uint32
   |  |  +--:(precentage)
   |  |     +--rw percent-value?    uint32
   |  +--rw (bc-model-type)?
   |     +--:(bc-model-rdm)
   |     |  +--rw bc-model-rdm
   |     |     +--rw bandwidth-psc-constraints
   |     |        +--rw maximum-reservable?   uint32
   |     |        +--rw bc-value*             uint32
```

```
   |     +--:(bc-model-mam)
   |     |  +--rw bc-model-mam
   |     |     +--rw bandwidth-psc-constraints
   |     |        +--rw maximum-reservable?   uint32
   |     |        +--rw bc-value*             uint32
   |     +--:(bc-model-mar)
   |        +--rw bc-model-mar
   |           +--rw bandwidth-psc-constraints
   |              +--rw maximum-reservable?   uint32
   |              +--rw bc-value*             uint32
   +--ro state
   |  +--ro (bandwidth-value)?
   |  |  +--:(absolute)
   |  |  |  +--ro absolute-value?                  uint32
   |  |  +--:(precentage)
   |  |     +--ro percent-value?                   uint32
   |  +--ro (bc-model-type)?
   |  |  +--:(bc-model-rdm)
   |  |  |  +--ro bc-model-rdm
   |  |  |     +--ro bandwidth-psc-constraints
   |  |  |        +--ro maximum-reservable?   uint32
   |  |  |        +--ro bc-value*             uint32
   |  |  +--:(bc-model-mam)
   |  |  |  +--ro bc-model-mam
   |  |  |     +--ro bandwidth-psc-constraints
   |  |  |        +--ro maximum-reservable?   uint32
   |  |  |        +--ro bc-value*             uint32
   |  |  +--:(bc-model-mar)
   |  |     +--ro bc-model-mar
   |  |        +--ro bandwidth-psc-constraints
   |  |           +--ro maximum-reservable?   uint32
   |  |           +--ro bc-value*             uint32
   |  +--ro interface-softpreemption-state
   |  |  +--ro soft-preempted-bandwidth?   uint32
   |  |  +--ro lsps* [source destination tunnel-id lsp-id extended-tunnel-id]
   |  |     +--ro source               -> /ietf-te:te/lsps-state/lsp/source
   |  |     +--ro destination          -> /ietf-te:te/lsps-state/lsp/
destination
   |  |     +--ro tunnel-id            -> /ietf-te:te/lsps-state/lsp/tunnel-id
   |  |     +--ro lsp-id               -> /ietf-te:te/lsps-state/lsp/lsp-id
   |  |     +--ro extended-tunnel-id   -> /ietf-te:te/lsps-state/lsp/extended-
tunnel-id
   |  |     +--ro type?                -> /ietf-te:te/lsps-state/lsp/type
   |  +--ro over-subscribed-bandwidth?        uint32
   +--rw rsvp-te-frr-backups {ietf-te-types:frr-te}?
      +--rw config
      |  +--rw backup-bandwidth?               uint32
      |  +--rw backup-bandwidth-classtype?     uint32
```

```
|  +--rw (type)?
|     +--:(static-tunnel)
```

```
      |     |  +--rw static-backups* [tunnel-name]
      |     |     +--rw tunnel-name    string
      |     +--:(auto-tunnel)
      |        +--rw auto-backup-protection?         identityref
      |        +--rw auto-backup-path-computation?   identityref
      +--ro state
         +--ro backup-bandwidth?               uint32
         +--ro backup-bandwidth-classtype?     uint32
         +--ro (type)?
            +--:(static-tunnel)
            |  +--ro static-backups* [tunnel-name]
            |     +--ro tunnel-name    string
            +--:(auto-tunnel)
               +--ro auto-backup-protection?         identityref
               +--ro auto-backup-path-computation?   identityref
augment /ietf-te:te/ietf-te:lsps-state/ietf-te:lsp:
   +--ro config
   |  +--ro static-bandwidth?              uint32
   |  +--ro auto-bandwidth
   |  |  +--ro enabled?            boolean
   |  |  +--ro min-bw?             uint32
   |  |  +--ro max-bw?             uint32
   |  |  +--ro adjust-interval?    uint32
   |  |  +--ro adjust-threshold?   uint32
   |  |  +--ro overflow
   |  |  |  +--ro enabled?               boolean
   |  |  |  +--ro overflow-threshold?    uint32
   |  |  |  +--ro trigger-event-count?   uint16
   |  |  +--ro underflow
   |  |     +--ro enabled?               boolean
   |  |     +--ro underflow-threshold?   uint32
   |  |     +--ro trigger-event-count?   uint16
   |  +--ro local-protection-desired?       empty
   |  +--ro bandwidth-protection-desired?   empty
   |  +--ro node-protection-desired?        empty
   |  +--ro non-php-desired?                empty
   |  +--ro entropy-label-cap?              empty
   |  +--ro oam-mep-entities-desired?       empty
   |  +--ro oam-mip-entities-desired?       empty
   +--ro state
   |  +--ro static-bandwidth?              uint32
   |  +--ro auto-bandwidth
   |  |  +--ro enabled?            boolean
   |  |  +--ro min-bw?             uint32
   |  |  +--ro max-bw?             uint32
   |  |  +--ro adjust-interval?    uint32
   |  |  +--ro adjust-threshold?   uint32
   |  |  +--ro overflow
```

```
|  |  |  +--ro enabled?                boolean
|  |  |  +--ro overflow-threshold?     uint32
|  |  |  +--ro trigger-event-count?    uint16
|  |  +--ro underflow
|  |     +--ro enabled?                boolean
|  |     +--ro underflow-threshold?    uint32
|  |     +--ro trigger-event-count?    uint16
|  +--ro local-protection-desired?        empty
|  +--ro bandwidth-protection-desired?    empty
|  +--ro node-protection-desired?         empty
|  +--ro non-php-desired?                 empty
|  +--ro entropy-label-cap?               empty
|  +--ro oam-mep-entities-desired?        empty
|  +--ro oam-mip-entities-desired?        empty
+--ro backup-info
   +--ro backup-tunnel-name?         string
   +--ro backup-frr-on?              uint8
   +--ro backup-protected-lsp-num?   uint32
```

3.4.2. RPC and Notification Data

TBD.

3.4.3. YANG Module

```
<CODE BEGINS> file "ietf-rsvp-te-psc@2015-10-16.yang"

module ietf-rsvp-te-psc {

  namespace "urn:ietf:params:xml:ns:yang:ietf-rsvp-te-psc";

  prefix "rsvp-te-psc";

  import ietf-rsvp {
    prefix rsvp;
  }

  import ietf-te {
    prefix ietf-te;
  }

  import ietf-te-psc-types {
    prefix ietf-te-psc-types;
  }

  import ietf-te-types {
    prefix ietf-te-types;
  }
```

```
     organization
       "IETF TEAS Working Group";

     contact "TBA";

     description
       "Latest update to RSVP-TE packet YANG data model.";

     revision 2015-10-16 {
       description "Update to RSVP-TE packet YANG initial revision.";
       reference "RFC3209, RFC6511, RFC6790, RFC7260, RFC4859, RFC4090";
     }

     /* RSVP-TE LSPs packet groupings */
     grouping lsp-attributes-flags-psc_config {
       description
         "Configuration parameters relating to RSVP-TE LSP
         packet attribute flags";
       leaf non-php-desired {
         type empty;
         description
           "Non-PHP is desired";
         reference "RFC6511";
       }
       leaf entropy-label-cap {
         type empty;
         description "Entropy label capability";
         reference "RFC6790";
       }
       leaf oam-mep-entities-desired {
         type empty;
         description "OAM MEP entities desired";
         reference "RFC7260";
       }
       leaf oam-mip-entities-desired {
         type empty;
         description "OAM MIP entities desired";
         reference "RFC7260";
       }
     }

     grouping lsp-session-attributes-obj-flags-psc_config {
       description
         "Configuration parameters relating to RSVP-TE LSP
         packet session attribute flags";
       reference
         "RFC4859: Registry for RSVP-TE Session Flags";
       leaf local-protection-desired {
```

```
        type empty;
        description "Fastreroute local protection is desired.";
        reference
          "RFC4859: Registry for RSVP-TE Session Flags";
      }
      leaf bandwidth-protection-desired {
        type empty;
        description
          "Request FRR bandwidth protection on LSRs if
          present.";
        reference "RFC4090";
      }
      leaf node-protection-desired {
        type empty;
        description
          "Request FRR node protection on LSRs if
          present.";
        reference "RFC4090";
      }
    }

    grouping  lsp-overflow_config {
      description "configuration for mpls lsp bandwidth
        overflow adjustment";

      leaf enabled {
        type boolean;
        default false;
        description "enables mpls lsp bandwidth overflow
          adjustment on the lsp";
      }

      leaf overflow-threshold {
        type uint32;
        description "bandwidth percentage change to trigger
          an overflow event";

      }

      leaf trigger-event-count {
        type uint16;
        description "number of consecutive overflow sample
          events needed to trigger an overflow adjustment";
      }
    }

    grouping lsp-underflow_config {
      description
```

```
        "configuration for mpls lsp bandwidth
        underflow adjustment";

      leaf enabled {
        type boolean;
        default false;
        description "enables bandwidth underflow
          adjustment on the lsp";
      }

      leaf underflow-threshold {
        type uint32;
        description "bandwidth percentage change to trigger
          and underflow event";
      }

      leaf trigger-event-count {
        type uint16;
        description "number of consecutive underflow sample
          events needed to trigger an underflow adjustment";
      }
    }

    grouping lsp-auto-bandwidth_config {
      description
        "Auto-Bandwidth grouping";
      container auto-bandwidth {
        description "configure auto-bandwidth operation in
          which devices automatically adjust bandwidth to meet
          requirements";

        leaf enabled {
          type boolean;
          default false;
          description "enables mpls auto-bandwidth on the
            lsp";
        }

        leaf min-bw {
          type uint32;
          description "set the minimum bandwidth in Mbps for an
            auto-bandwidth LSP";
        }

        leaf max-bw {
          type uint32;
          description "set the maximum bandwidth in Mbps for an
            auto-bandwidth LSP";
```

```
        }

        leaf adjust-interval {
          type uint32;
          description "time in seconds between adjustments to
            LSP bandwidth";
        }

        leaf adjust-threshold {
          type uint32;
          description "percentage difference between the LSP's
            specified bandwidth and its current bandwidth
            allocation -- if the difference is greater than the
            specified percentage, auto-bandwidth adjustment is
            triggered";
        }

        container overflow {
          description "configuration of MPLS overflow bandwidth
            adjustement for the LSP";
          uses lsp-overflow_config;
        }

        container underflow {
          description "configuration of MPLS underflow bandwidth
            adjustement for the LSP";
          uses lsp-underflow_config;
        }
      }
    }

    grouping lsp-bandwidth_config {
      description
        "LSP bandwidth grouping";
      leaf static-bandwidth {
        type uint32;
        description
          "Static bandwidth, e.g., using
           offline calculation";
      }
      uses lsp-auto-bandwidth_config;
    }

    grouping lsp-properties-psc {
      description
        "Top level grouping for LSP properties.";
      container config {
        description
```

```
          "Configuration parameters relating to
          LSP properties";
        uses lsp-bandwidth_config;
        uses lsp-session-attributes-obj-flags-psc_config;
        uses lsp-attributes-flags-psc_config;
      }
      container state {
        config false;
        description
          "State information associated with LSP
          properties";
        uses lsp-bandwidth_config;
        uses lsp-session-attributes-obj-flags-psc_config;
        uses lsp-attributes-flags-psc_config;
      }
    }
    /* End of RSVP-TE LSPs packet groupings */

    /* RSVP-TE packet interface groupings */
    grouping rsvp-te-interface_state {
      description
        "The RSVP-TE interface state grouping";
      leaf over-subscribed-bandwidth {
        type uint32;
        description
          "The amount of over-subscribed bandwidth on
          the interface";
      }
    }

    grouping rsvp-te-interface-softpreemption_state {
      description
        "The RSVP-TE interface preeemptions state grouping";
      container interface-softpreemption-state {
        description
          "The RSVP-TE interface preeemptions state grouping";
        leaf soft-preempted-bandwidth {
          type uint32;
          description
            "The amount of soft-preempted bandwidth on
            this interface";
        }
        list lsps {
          key
            "source destination tunnel-id lsp-id "+
            "extended-tunnel-id";
          description
            "List of LSPs that are soft-preempted";
```

```
            leaf source {
              type leafref {
                path "/ietf-te:te/ietf-te:lsps-state/ietf-te:lsp/"+
                     "ietf-te:source";
              }
              description
                "Tunnel sender address extracted from
                SENDER_TEMPLATE  object";
              reference "RFC3209";
            }
            leaf destination {
              type leafref {
                path "/ietf-te:te/ietf-te:lsps-state/ietf-te:lsp/"+
                     "ietf-te:destination";
              }
              description
                "Tunnel endpoint address extracted from
                SESSION object";
              reference "RFC3209";
            }
            leaf tunnel-id {
              type leafref {
                path "/ietf-te:te/ietf-te:lsps-state/ietf-te:lsp/"+
                  "ietf-te:tunnel-id";
              }
              description
                "Tunnel identifier used in the SESSION
                that remains constant over the life
                of the tunnel.";
              reference "RFC3209";
            }
            leaf lsp-id {
              type leafref {
                path "/ietf-te:te/ietf-te:lsps-state/ietf-te:lsp/"+
                     "ietf-te:lsp-id";
              }
              description
                "Identifier used in the SENDER_TEMPLATE
                and the FILTER_SPEC that can be changed
                to allow a sender to share resources with
                itself.";
              reference "RFC3209";
            }
            leaf extended-tunnel-id {
              type leafref {
                path "/ietf-te:te/ietf-te:lsps-state/ietf-te:lsp/"+
                  "ietf-te:extended-tunnel-id";
              }
```

```
            description
              "Extended Tunnel ID of the LSP.";
            reference "RFC3209";
          }
          leaf type {
            type leafref {
              path "/ietf-te:te/ietf-te:lsps-state/ietf-te:lsp/"+
                "ietf-te:type";
            }
            description "LSP type P2P or P2MP";
          }
        }
      }
    }
    /* End of RSVP-TE interface groupings */

    /* RSVP-TE FRR groupings */
    grouping rsvp-te-frr-backups_config {
      description
        "Top level container for RSVP-TE FRR backup parameters";
      leaf backup-bandwidth {
        type uint32;
        description
          "Maximum bandwidth this facility backup
          is allowed to protect";
      }
      leaf backup-bandwidth-classtype {
        type uint32;
        description
          "Type of primary LSP bandwidth that the
          backup is allowed to protect.";
      }
      choice type {
        description
          "FRR backup tunnel type";
        case static-tunnel {
          list static-backups {
            key "tunnel-name";
            description
              "List of static backup tunnels that
              protect the RSVP-TE interface.";
            leaf tunnel-name {
              type string;
              description "FRR Backup tunnel";
            }
          }
        }
        case auto-tunnel {
```

```
          leaf auto-backup-protection {
            type identityref {
              base ietf-te-psc-types:backup-protection-type;
            }
            default
              ietf-te-psc-types:backup-protection-node-link;
            description
              "Describes whether the backup should offer
              protection against link, node, or either";
          }
          leaf auto-backup-path-computation {
            type identityref {
              base
                ietf-te-types:path-computation-srlg-type;
            }
            description
              "FRR backup computation type";
          }
        }
      }
    }

    grouping rsvp-te-frr-backups {
      description
        "Top level grouping for RSVP-TE FRR backup properties.";
      container rsvp-te-frr-backups {
        if-feature ietf-te-types:frr-te;
        description
          "Top level container for RSVP-TE FRR backup
          properties.";
        container config {
          description
            "Configuration parameters for interface RSVP-TE
            FRR backup properties";
          uses rsvp-te-frr-backups_config;
        }
        container state {
          config false;
          description
            "State parameters for interface RSVP-TE
            FRR backup properties";
          uses rsvp-te-frr-backups_config;
        }
      }
    }

    grouping lps-backup-info_state {
      description "Backup/bypass LSP related information";
```

```
       container backup-info {
         description
           "backup information";

         leaf backup-tunnel-name {
           type string;
           description
             "If an LSP has an FRR backup LSP that can protect it,
             this field identifies the tunnel name of the backup LSP.
             Otherwise, this field is empty.";
         }

         leaf backup-frr-on {
           type uint8;
           description
             "Whether currently this backup is carrying traffic";
         }

         leaf backup-protected-lsp-num {
           type uint32;
           description
             "Number of LSPs protected by this backup";
         }
       }
     }

     grouping rsvp-frr-local-revert_config {
       description "RSVP-TE FRR local revertive grouping";
       container rsvp-frr-local-revert {
         presence "Enable RSVP FRR local revertive recovery
           mode.";
         description
           "RSVP-TE global properties container";
         leaf rsvp-frr-local-revert-delay {
           type uint32;
           description
             "Time to wait after primary link is restored
             before node attempts local revertive
             procedures.";
         }
       }
     }

     /*** End of RSVP-TE FRR backup information ***/

     grouping globals-properties {
       description
         "Top level grouping for globals properties";
```

```
      container config {
        description
          "Configuration parameters relating to
           global RVP-TE packet properties";
        uses rsvp-frr-local-revert_config;
      }
      container state {
        config false;
        description
          "State parameters relating to
           global RVP-TE packet properties";
        uses rsvp-frr-local-revert_config;
      }
    }

    /* RSVP-TE global propeerties */
    augment "/rsvp:rsvp/rsvp:globals" {
      description
        "RSVP-TE augmentation to RSVP globals";
      uses globals-properties;
    }

    grouping rsvp-te-interface-attributes-psc {
      description
        "Top level grouping for RSVP-TE packet interface
        properties.";
      container config {
        description
          "Configuration parameters relating to RSVP-TE
          bandwidth";
        uses ietf-te-psc-types:bandwidth-psc-reservable;
      }
      container state {
        config false;
        description
          "State information associated with RSVP-TE
          bandwidth";
        uses ietf-te-psc-types:bandwidth-psc-reservable;
        uses rsvp-te-interface-softpreemption_state;
        uses rsvp-te-interface_state;
      }
    }

    /* Linkage to the base RSVP all links */
    augment "/rsvp:rsvp/rsvp:interfaces" {
      description "TBD";
      /* To be added */
    }
```

```
    /* Linkage to per RSVP link */
    augment "/rsvp:rsvp/rsvp:interfaces/rsvp:interface" {
      description "TBD";
      /* To be added */
    }

    /* add augmentation for sessions neighbors */
    augment "/rsvp:rsvp/rsvp:sessions" {
      description "TBD";
      /* To be added */
    }

    augment "/rsvp:rsvp/rsvp:neighbors" {
      description "TBD";
      /* To be added */
    }

    /**
     * Augmentation to TE generic module
     */
    augment "/ietf-te:te/ietf-te:tunnels/ietf-te:tunnel" {
      description "TBD";
      uses lsp-properties-psc;
    }

    augment "/ietf-te:te/ietf-te:interfaces/ietf-te:interface" {
      description
        "RSVP reservable bandwidth configuration properties";
      uses rsvp-te-interface-attributes-psc;
      uses rsvp-te-frr-backups;
    }

    augment "/ietf-te:te/ietf-te:lsps-state/ietf-te:lsp" {
      description
        "RSVP-TE LSP state properties";
      uses lsp-properties-psc;
      uses lps-backup-info_state;
    }
  }

  <CODE ENDS>
```

4. IANA Considerations

This document registers the following URIs in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested to be made.

URI: urn:ietf:params:xml:ns:yang:ietf-rsvp XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-rsvp-te XML: N/A, the requested URI is an XML namespace.

This document registers a YANG module in the YANG Module Names registry [RFC6020].

name: ietf-rsvp namespace: urn:ietf:params:xml:ns:yang:ietf-rsvp prefix: ietf-rsvp reference: RFC3209

name: ietf-rsvp-te namespace: urn:ietf:params:xml:ns:yang:ietf-rsvp-te prefix: ietf-rsvp-te reference: RFC3209

5. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242]. The NETCONF access control model [RFC6536] provides means to restrict access for particular NETCONF

users to a pre-configured subset of all available NETCONF protocol operations and content.

There are a number of data nodes defined in the YANG module which are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., <edit-config>) to these data nodes without proper protection can have a negative effect on network operations.

6. Acknowledgement

The authors would like to thank Lou Berger for reviewing and providing valuable feedback on this document.

7. References

7.1. Normative References

[I-D.ietf-netmod-routing-cfg]
Lhotka, L. and A. Lindem, "A YANG Data Model for Routing Management", draft-ietf-netmod-routing-cfg-20 (work in progress), October 2015.

[I-D.ietf-teas-yang-te]
Saad, T., Gandhi, R., Liu, X., Beeram, V., Shah, H., Chen, X., Jones, R., and B. Wen, "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te-00 (work in progress), September 2015.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <http://www.rfc-editor.org/info/rfc2205>.

[RFC2747] Baker, F., Lindell, B., and M. Talwar, "RSVP Cryptographic Authentication", RFC 2747, DOI 10.17487/RFC2747, January 2000, <http://www.rfc-editor.org/info/rfc2747>.

[RFC2961] Berger, L., Gan, D., Swallow, G., Pan, P., Tommasi, F., and S. Molendini, "RSVP Refresh Overhead Reduction Extensions", RFC 2961, DOI 10.17487/RFC2961, April 2001, <http://www.rfc-editor.org/info/rfc2961>.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <http://www.rfc-editor.org/info/rfc3209>.

[RFC3473] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, DOI 10.17487/RFC3473, January 2003, <http://www.rfc-editor.org/info/rfc3473>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <http://www.rfc-editor.org/info/rfc3688>.

[RFC5063] Satyanarayana, A., Ed. and R. Rahman, Ed., "Extensions to GMPLS Resource Reservation Protocol (RSVP) Graceful Restart", RFC 5063, DOI 10.17487/RFC5063, October 2007, <http://www.rfc-editor.org/info/rfc5063>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <http://www.rfc-editor.org/info/rfc6020>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <http://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <http://www.rfc-editor.org/info/rfc6242>.

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <http://www.rfc-editor.org/info/rfc6536>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <http://www.rfc-editor.org/info/rfc6991>.

7.2. Informative References

[I-D.openconfig-mpls-consolidated-model]
George, J., Fang, L., eric.osborne@level3.com, e., and R. Shakir, "MPLS / TE Model for Service Provider Networks", draft-openconfig-mpls-consolidated-model-01 (work in progress), July 2015.

[I-D.openconfig-netmod-opstate]
Shakir, R., Shaikh, A., and M. Hines, "Consistent Modeling of Operational State Data in YANG", draft-openconfig-netmod-opstate-01 (work in progress), July 2015.

Authors' Addresses

Vishnu Pavan Beeram
Juniper Networks

Email: vbeeram@juniper.net

Tarek Saad
Cisco Systems Inc

Email: tsaad@cisco.com

Rakesh Gandhi
Cisco Systems Inc

Email: rgandhi@cisco.com

Xufeng Liu
Ericsson

Email: xufeng.liu@ericsson.com

Himanshu Shah
Ciena

Email: tsaad@cisco.com

Xia Chen
Huawei Technologies

Email: jescia.chenxia@huawei.com

Raqib Jones
Brocade

Email: raqib@Brocade.com

Bin Wen
Comcast

Email: Bin_Wen@cable.comcast.com