

TEAS Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 28, 2021

V. Beeram  
T. Saad  
Juniper Networks  
R. Gandhi  
Cisco Systems, Inc.  
X. Liu  
Volta Networks  
I. Bryskin  
Individual  
July 27, 2020

**A YANG Data Model for Resource Reservation Protocol (RSVP)**  
**draft-ietf-teas-yang-rsvp-14**

Abstract

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

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 <https://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 January 28, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|                       |                                       |                    |
|-----------------------|---------------------------------------|--------------------|
| <a href="#">1.</a>    | Introduction . . . . .                | <a href="#">2</a>  |
| <a href="#">2.</a>    | Requirements Language . . . . .       | <a href="#">3</a>  |
| <a href="#">2.1.</a>  | Prefixes in Data Node Names . . . . . | <a href="#">3</a>  |
| <a href="#">2.2.</a>  | Model Tree Diagram . . . . .          | <a href="#">4</a>  |
| <a href="#">3.</a>    | Model Overview . . . . .              | <a href="#">4</a>  |
| <a href="#">3.1.</a>  | Module(s) Relationship . . . . .      | <a href="#">4</a>  |
| <a href="#">3.2.</a>  | Design Considerations . . . . .       | <a href="#">5</a>  |
| <a href="#">3.3.</a>  | Model Notifications . . . . .         | <a href="#">6</a>  |
| <a href="#">4.</a>    | RSVP Base YANG Model . . . . .        | <a href="#">6</a>  |
| <a href="#">4.1.</a>  | Module Structure . . . . .            | <a href="#">6</a>  |
| <a href="#">4.2.</a>  | Tree Diagram . . . . .                | <a href="#">8</a>  |
| <a href="#">4.3.</a>  | YANG Module . . . . .                 | <a href="#">13</a> |
| <a href="#">5.</a>    | RSVP Extended YANG Model . . . . .    | <a href="#">34</a> |
| <a href="#">5.1.</a>  | Module Structure . . . . .            | <a href="#">34</a> |
| <a href="#">5.2.</a>  | Tree Diagram . . . . .                | <a href="#">34</a> |
| <a href="#">5.3.</a>  | YANG Module . . . . .                 | <a href="#">36</a> |
| <a href="#">6.</a>    | IANA Considerations . . . . .         | <a href="#">44</a> |
| <a href="#">7.</a>    | Security Considerations . . . . .     | <a href="#">45</a> |
| <a href="#">8.</a>    | Acknowledgement . . . . .             | <a href="#">46</a> |
| <a href="#">9.</a>    | Contributors . . . . .                | <a href="#">46</a> |
| <a href="#">10.</a>   | References . . . . .                  | <a href="#">47</a> |
| <a href="#">10.1.</a> | Normative References . . . . .        | <a href="#">47</a> |
| <a href="#">10.2.</a> | Informative References . . . . .      | <a href="#">49</a> |
|                       | Authors' Addresses . . . . .          | <a href="#">50</a> |

## [1.](#) Introduction

YANG [[RFC6020](#)] and [[RFC7950](#)] is a data modeling language that was introduced to define the contents of a conceptual data store that allows networked devices to be managed using NETCONF [[RFC6241](#)]. YANG has proved relevant beyond its initial confines, as bindings to other interfaces (e.g. RESTCONF [[RFC8040](#)]) 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 interfaces, such as CLI and programmatic APIs.



This document defines a YANG data model that can be used to configure and manage the RSVP protocol [RFC2205]. The model is separated into two modules: a base and RSVP extended YANG modules. The RSVP base YANG module models the data that is core to the function of the RSVP protocol and MUST be supported by vendors that support RSVP protocol [RFC2205]. The RSVP extended module models data that is either optional or provides ability to tune basic RSVP protocol functionality. The support for RSVP extended features by all vendors is considered optional.

The RSVP YANG model provides the building blocks needed to allow augmentation by other models that extend the RSVP protocol- such as using RSVP extensions to signal Label Switched Paths (LSPs) as defined in [RFC3209].

The YANG module(s) defined in this document are compatible with the Network Management Datastore Architecture (NMDA) [RFC7950].

## 2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The terminology for describing YANG data models is found in [RFC7950].

### 2.1. 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] |
| rt-types  | ietf-routing-types | [RFC8294] |
| key-chain | ietf-key-chain     | [RFC8177] |

Table 1: Prefixes and corresponding YANG modules



## **2.2. Model Tree Diagram**

A full tree diagram of the module(s) defined in this document is given in subsequent sections as per the syntax defined in [[RFC8340](#)].

## **3. Model Overview**

The RSVP base YANG module augments the "control-plane-protocol" list in ietf-routing [[RFC8349](#)] module with specific RSVP parameters in an "rsvp" container. It also defines an extension identity "rsvp" of base "rt:routing-protocol" to identify the RSVP protocol.

The augmentation of the RSVP model by other models (e.g. YANG models that support RSVP Traffic Engineering (TE) extensions for signaling Label Switched Paths (LSPs)) are outside the scope of this document and are discussed in separate document(s), e.g. [[I-D.ietf-teas-yang-rsvp-te](#)].

### **3.1. Module(s) Relationship**

This document divides the RSVP model into two modules: base and RSVP extended modules. Some RSVP data are categorized as core to the function of the protocol and MUST be supported by vendors claiming the support for RSVP protocol [[RFC2205](#)]. Such configuration and state data are grouped in the RSVP base module.

Other RSVP extended features are categorized as either optional or providing ability to better tune the basic functionality of the RSVP protocol. The support for RSVP extended features by all vendors is considered optional. Such features are grouped in a separate RSVP extended module.

The relationship between the base and RSVP extended YANG modules and the IETF routing YANG model is shown in Figure 1.



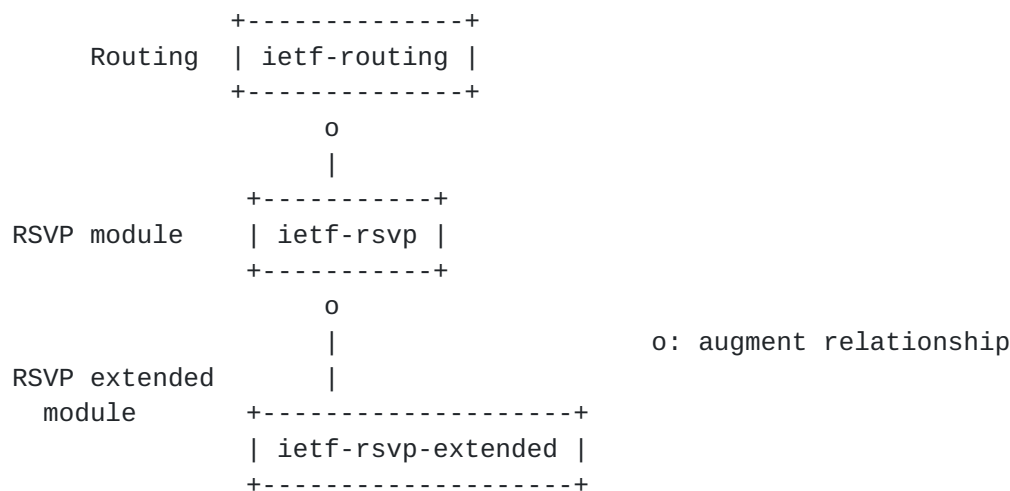


Figure 1: Relationship of RSVP and RSVP extended modules with other protocol modules

### 3.2. Design Considerations

The RSVP base model does not aim to be feature complete. The primary intent is to cover a set of standard core features that are commonly in use. For example:

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

The RSVP extended YANG module covers the configuration for optional features that are not must for basic RSVP protocol operation.

The defined data model supports configuration inheritance for neighbors, and interfaces. Data nodes defined under 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.



### **3.3. Model Notifications**

Notifications data modeling is key in any defined data model.

[RFC8639] and [[RFC8641](#)] define a subscription and push mechanism for YANG datastores. This mechanism currently allows the user to:

- o Subscribe notifications on a per client basis
- o Specify subtree filters or xpath filters so that only interested contents will be sent.
- o Specify either periodic or on-demand notifications.

## **4. RSVP Base YANG Model**

The RSVP base module defines the main building blocks for modeling the RSVP protocol and augments the IETF routing module.

### **4.1. Module Structure**

The RSVP base YANG data model defines the container "rsvp" as the top level container in this data model. The presence of this container enables the RSVP protocol functionality.

The derived state data is contained in "read-only" nodes directly under the intended object as shown in Figure 2.



```
module: ietf-rsvp
  +--rw rsvp!
    +--rw globals
      .
      .
    +--rw interfaces
      .
      +-- ro <<derived state associated with interfaces>>
      .
      .
    +--rw neighbors
      .
      +-- ro <<derived state associated with the LSP Tunnel>>
      .
      .
    +--rw sessions
      .
      +-- ro <<derived state associated with the LSP Tunnel>>
      .
  rpcs:
    +--x clear-session
    +--x clear-neighbor
    +--x clear-authentication
```

Figure 2: RSVP high-level tree model view

Configuration and state data are grouped to those applicable on per node (global), per interface, per neighbor, or per session.

'globals':

The globals container includes configuration and state data that is applicable globally and affects the RSVP protocol behavior.

'interfaces':

The 'interfaces' container includes a list of RSVP enabled interfaces. It also includes configuration and state data that are applicable to all interfaces. An entry in the interfaces list MAY carry its own configuration or state data. Any data or state under the "interfaces" container level is equally applicable to all interfaces unless it is explicitly overridden by configuration or state under a specific interface.

'neighbors' :



The 'neighbors' container includes a list of RSVP neighbors. An entry in the RSVP neighbor list MAY carry its own configuration and state relevant to the specific RSVP neighbor. RSVP neighbors can be dynamically discovered using RSVP signaling or explicitly configured.

'sessions':

The 'sessions' container includes a list RSVP sessions. An entry in the RSVP session list MAY carry its own configuration and state relevant to a specific RSVP session. RSVP sessions are usually derived state that are created as result of signaling. This model defines attributes related to IP RSVP sessions as defined in [\[RFC2205\]](#).

#### 4.2. Tree Diagram

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

```

module: ietf-rsvp
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol:
      +--rw rsvp!
        +--rw globals
          | +--rw sessions
          | | +--ro session-ip*
          | | [destination protocol-id destination-port]
          | | +--ro destination-port    inet:port-number
          | | +--ro protocol-id         uint8
          | | +--ro source?             inet:ip-address
          | | +--ro destination         inet:ip-address
          | | +--ro session-name?       string
          | | +--ro session-status?     enumeration
          | | +--ro session-type?       identityref
          | | +--ro psbs
          | | | +--ro psb* []
          | | | | +--ro source-port?    inet:port-number
          | | | | +--ro expires-in?     uint32
          | | +--ro rsbs
          | | | +--ro rsb* []
          | | | | +--ro source-port?     inet:port-number
          | | | | +--ro reservation-style? identityref
          | | | | +--ro expires-in?      uint32
          | +--ro statistics
          | | +--ro messages
          | | | +--ro ack-sent?          yang:counter64
          | | | +--ro ack-received?     yang:counter64
  
```



```

| | | +--ro bundle-sent?                yang:counter64
| | | +--ro bundle-received?            yang:counter64
| | | +--ro hello-sent?                 yang:counter64
| | | +--ro hello-received?             yang:counter64
| | | +--ro integrity-challenge-sent?    yang:counter64
| | | +--ro integrity-challenge-received? yang:counter64
| | | +--ro integrity-response-sent?     yang:counter64
| | | +--ro integrity-response-received? yang:counter64
| | | +--ro notify-sent?                 yang:counter64
| | | +--ro notify-received?             yang:counter64
| | | +--ro path-sent?                   yang:counter64
| | | +--ro path-received?               yang:counter64
| | | +--ro path-err-sent?               yang:counter64
| | | +--ro path-err-received?           yang:counter64
| | | +--ro path-tear-sent?              yang:counter64
| | | +--ro path-tear-received?          yang:counter64
| | | +--ro resv-sent?                   yang:counter64
| | | +--ro resv-received?               yang:counter64
| | | +--ro resv-confirm-sent?           yang:counter64
| | | +--ro resv-confirm-received?       yang:counter64
| | | +--ro resv-err-sent?               yang:counter64
| | | +--ro resv-err-received?           yang:counter64
| | | +--ro resv-tear-sent?              yang:counter64
| | | +--ro resv-tear-received?          yang:counter64
| | | +--ro summary-refresh-sent?        yang:counter64
| | | +--ro summary-refresh-received?    yang:counter64
| | | +--ro unknown-messages-received?   yang:counter64
| | +--ro packets
| | | +--ro sent?                        yang:counter64
| | | +--ro received?                   yang:counter64
| | +--ro errors
| | | +--ro authenticate?               yang:counter64
| | | +--ro checksum?                   yang:counter64
| | | +--ro packet-length?               yang:counter64
| +--rw graceful-restart
|   +--rw enabled?                       boolean
+--rw interfaces
| +--rw refresh-reduction
| | +--rw enabled?                       boolean
| +--rw hellos
| | +--rw enabled?                       boolean
| +--rw authentication
| | +--rw enabled?                       boolean
| | +--rw authentication-key?            string
| | +--rw crypto-algorithm               identityref
| +--ro statistics
| | +--ro messages
| | | +--ro ack-sent?                    yang:counter64

```



```

| | | +--ro ack-received?                yang:counter64
| | | +--ro bundle-sent?                yang:counter64
| | | +--ro bundle-received?            yang:counter64
| | | +--ro hello-sent?                 yang:counter64
| | | +--ro hello-received?             yang:counter64
| | | +--ro integrity-challenge-sent?    yang:counter64
| | | +--ro integrity-challenge-received? yang:counter64
| | | +--ro integrity-response-sent?     yang:counter64
| | | +--ro integrity-response-received? yang:counter64
| | | +--ro notify-sent?                 yang:counter64
| | | +--ro notify-received?             yang:counter64
| | | +--ro path-sent?                   yang:counter64
| | | +--ro path-received?               yang:counter64
| | | +--ro path-err-sent?               yang:counter64
| | | +--ro path-err-received?           yang:counter64
| | | +--ro path-tear-sent?              yang:counter64
| | | +--ro path-tear-received?          yang:counter64
| | | +--ro resv-sent?                   yang:counter64
| | | +--ro resv-received?               yang:counter64
| | | +--ro resv-confirm-sent?           yang:counter64
| | | +--ro resv-confirm-received?       yang:counter64
| | | +--ro resv-err-sent?               yang:counter64
| | | +--ro resv-err-received?           yang:counter64
| | | +--ro resv-tear-sent?              yang:counter64
| | | +--ro resv-tear-received?          yang:counter64
| | | +--ro summary-refresh-sent?        yang:counter64
| | | +--ro summary-refresh-received?    yang:counter64
| | | +--ro unknown-messages-received?   yang:counter64
| | +--ro packets
| | | +--ro sent?                yang:counter64
| | | +--ro received?            yang:counter64
| | +--ro errors
| | | +--ro authenticate?        yang:counter64
| | | +--ro checksum?            yang:counter64
| | | +--ro packet-length?        yang:counter64
+--rw interface* [interface]
|   +--rw interface                if:interface-ref
|   +--rw refresh-reduction
|   | +--rw enabled?              boolean
|   +--rw hellos
|   | +--rw enabled?              boolean
|   +--rw authentication
|   | +--rw enabled?              boolean
|   | +--rw authentication-key?   string
|   | +--rw crypto-algorithm      identityref
|   +--ro statistics
|   | +--ro messages
|   | | +--ro ack-sent?

```



```
| | | yang:counter64
| | +--ro ack-received?
| | | yang:counter64
| | +--ro bundle-sent?
| | | yang:counter64
| | +--ro bundle-received?
| | | yang:counter64
| | +--ro hello-sent?
| | | yang:counter64
| | +--ro hello-received?
| | | yang:counter64
| | +--ro integrity-challenge-sent?
| | | yang:counter64
| | +--ro integrity-challenge-received?
| | | yang:counter64
| | +--ro integrity-response-sent?
| | | yang:counter64
| | +--ro integrity-response-received?
| | | yang:counter64
| | +--ro notify-sent?
| | | yang:counter64
| | +--ro notify-received?
| | | yang:counter64
| | +--ro path-sent?
| | | yang:counter64
| | +--ro path-received?
| | | yang:counter64
| | +--ro path-err-sent?
| | | yang:counter64
| | +--ro path-err-received?
| | | yang:counter64
| | +--ro path-tear-sent?
| | | yang:counter64
| | +--ro path-tear-received?
| | | yang:counter64
| | +--ro resv-sent?
| | | yang:counter64
| | +--ro resv-received?
| | | yang:counter64
| | +--ro resv-confirm-sent?
| | | yang:counter64
| | +--ro resv-confirm-received?
| | | yang:counter64
| | +--ro resv-err-sent?
| | | yang:counter64
| | +--ro resv-err-received?
| | | yang:counter64
| | +--ro resv-tear-sent?
```



```

| | yang:counter64
| | +--ro resv-tear-received?
| | | yang:counter64
| | +--ro summary-refresh-sent?
| | | yang:counter64
| | +--ro summary-refresh-received?
| | | yang:counter64
| | +--ro unknown-messages-received?
| | | yang:counter64
| +--ro packets
| | +--ro sent? yang:counter64
| | +--ro received? yang:counter64
| +--ro errors
| | +--ro authenticate? yang:counter64
| | +--ro checksum? yang:counter64
| | +--ro packet-length? yang:counter64
+--rw neighbors
  +--rw neighbor* [address]
    +--rw address inet:ip-address
    +--rw epoch? uint32
    +--rw expiry-time? uint32
    +--rw graceful-restart
    | +--rw enabled? boolean
    | +--rw local-restart-time? uint32
    | +--rw local-recovery-time? uint32
    | +--rw neighbor-restart-time? uint32
    | +--rw neighbor-recovery-time? uint32
    | +--rw helper-mode
    | | +--rw enabled? boolean
    | | +--rw max-helper-restart-time? uint32
    | | +--rw max-helper-recovery-time? uint32
    | | +--rw neighbor-restart-time-remaining? uint32
    | | +--rw neighbor-recovery-time-remaining? uint32
    +--rw hello-status? enumeration
    +--rw interface? if:interface-ref
    +--rw neighbor-status? enumeration
    +--rw refresh-reduction-capable? boolean
    +--rw restart-count? yang:counter32
    +--rw restart-time? yang:date-and-time

rpccs:
+---x clear-session
| +---w input
| | +---w routing-protocol-instance-name leafref
| | +---w (filter-type)
| | | +---:(match-all)
| | | | +---w all empty
| | | +---:(match-one)

```



```

|           +---w session-info
|           +---w (session-type)
|           +--:(rsvp-session-ip)
|           +---w destination          leafref
|           +---w protocol-id          uint8
|           +---w destination-port     inet:ip-address
+---x clear-neighbor
|   +---w input
|   +---w routing-protocol-instance-name leafref
|   +---w (filter-type)
|   +--:(match-all)
|   |   +---w all                      empty
|   +--:(match-one)
|   +---w neighbor-address            leafref
+---x clear-authentication
|   +---w input
|   +---w routing-protocol-instance-name leafref
|   +---w (filter-type)
|   +--:(match-all)
|   |   +---w all                      empty
|   +--:(match-one-interface)
|   +---w interface?                  if:interface-ref

```

Figure 3: RSVP model tree diagram

#### 4.3. YANG Module

The ietf-rsvp module imports from the following modules:

- o ietf-interfaces defined in [[RFC8343](#)]
- o ietf-yang-types and ietf-inet-types defined in [[RFC6991](#)]
- o ietf-routing defined in [[RFC8349](#)]
- o ietf-key-chain defined in [[RFC8177](#)]

This module references the following documents: [[RFC2205](#)], [[RFC2747](#)], and [[RFC2961](#)].

```

<CODE BEGINS> file "ietf-rsvp@2020-07-24.yang"
module ietf-rsvp {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-rsvp";

  /* Replace with IANA when assigned */

  prefix rsvp;

```



```
import ietf-interfaces {
  prefix if;
  reference
    "RFC8343: A YANG Data Model for Interface Management";
}
import ietf-inet-types {
  prefix inet;
  reference
    "RFC6991: Common YANG Data Types";
}
import ietf-yang-types {
  prefix yang;
  reference
    "RFC6991: Common YANG Data Types";
}
import ietf-routing {
  prefix rt;
  reference
    "RFC8349: A YANG Data Model for Routing Management
      (NMDA Version)";
}
import ietf-key-chain {
  prefix key-chain;
  reference
    "RFC8177: YANG Data Model for Key Chains";
}
import ietf-netconf-acm {
  prefix nacm;
  reference
    "RFC8341: Network Configuration Access Control Model";
}
organization
  "IETF Traffic Engineering Architecture and Signaling (TEAS)
  Working Group";
contact
  "WG Web:  <http://tools.ietf.org/wg/teas/>
  WG List:  <mailto:teas@ietf.org>

  Editor:   Vishnu Pavan Beeram
            <mailto:vbeeram@juniper.net>

  Editor:   Tarek Saad
            <mailto:tsaad@juniper.net>

  Editor:   Rakesh Gandhi
            <mailto:rgandhi@cisco.com>

  Editor:   Xufeng Liu
```



<mailto: xufeng.liu.ietf@gmail.com>

Editor: Igor Bryskin  
<mailto:i\_bryskin@yahoo.com>;

description

"This module contains the RSVP YANG data model.  
The model fully conforms to the Network Management Datastore  
Architecture (NMDA).

Copyright (c) 2019 IETF Trust and the persons  
identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or  
without modification, is permitted pursuant to, and subject  
to the license terms contained in, the Simplified BSD License  
set forth in [Section 4.c](https://trustee.ietf.org/license-info) of the IETF Trust's Legal Provisions  
Relating to IETF Documents  
(<https://trustee.ietf.org/license-info>).  
This version of this YANG module is part of RFC XXXX; see  
the RFC itself for full legal notices.";

// RFC Ed.: replace XXXX with actual RFC number and remove this  
// note.  
// RFC Ed.: update the date below with the date of RFC publication  
// and remove this note.

```
revision 2020-07-24 {  
  description  
    "Initial version.";  
  reference  
    "RFCXXXX: A YANG Data Model for Resource Reservation Protocol  
    (RSVP)";  
}
```

```
identity rsvp {  
  base rt:routing-protocol;  
  description  
    "RSVP protocol";  
}
```

```
identity rsvp-session-type {  
  description  
    "Base RSVP session type";  
}
```

```
identity rsvp-session-ip {  
  base rsvp-session-type;  
  description
```



```
    "RSVP IP session type";
}

identity reservation-style {
    description
        "Base identity for reservation style.";
}

identity reservation-wildcard-filter {
    base reservation-style;
    description
        "Wildcard-Filter (WF) Style.";
    reference
        "RFC2205";
}

identity reservation-fixed-filter {
    base reservation-style;
    description
        "Fixed-Filter (FF) Style.";
    reference
        "RFC2205";
}

identity reservation-shared-explicit {
    base reservation-style;
    description
        "Shared Explicit (SE) Style.";
    reference
        "RFC2205";
}

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
        "RSVP graceful restart parameters grouping.";
    container graceful-restart {
```



```
        description
            "RSVP graceful restart parameters container.";
        uses graceful-restart-config;
    }
}

grouping refresh-reduction-config {
    description
        "Configuration parameters related 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 Authentication is enabled.
            'false' if RSVP Authentication is disabled.";
    }
    leaf authentication-key {
        type string;
        description
            "An authentication key string.";
        reference
            "RFC2747: RSVP Cryptographic Authentication";
    }
    leaf crypto-algorithm {
        type identityref {
            base key-chain:crypto-algorithm;
        }
        mandatory true;
    }
}
```



```
        description
            "Cryptographic algorithm associated with key.";
    }
}

grouping authentication {
    description
        "Top level grouping for RSVP authentication parameters.";
    container authentication {
        description
            "Top level container for RSVP authentication parameters.";
        uses authentication-config;
    }
}

grouping 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 hellos {
    description
        "Top level grouping for RSVP hellos parameters.";
    container hellos {
        description
            "Top level container for RSVP hello parameters.";
        uses 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.";
leaf destination-port {
  type inet:port-number;
  description
    "RSVP destination port.";
  reference
    "RFC2205";
}
leaf protocol-id {
  type uint8;
  description
    "The IP protocol ID.";
  reference
    "RFC2205, section 3.2";
}
leaf source {
  type inet:ip-address;
  description
    "RSVP source address.";
  reference
    "RFC2205";
}
leaf destination {
  type inet:ip-address;
  description
    "RSVP destination address.";
  reference
    "RFC2205";
}
leaf session-name {
  type string;
  description
    "The signaled name of this RSVP session.";
}
leaf session-status {
  type enumeration {
    enum up {
      description
        "RSVP session is up.";
    }
    enum down {
      description
        "RSVP session is down.";
    }
  }
}
description
  "Enumeration of RSVP session states.";
```



```
}
leaf session-type {
  type identityref {
    base rsvp-session-type;
  }
  description
    "RSVP session type.";
}
container psbs {
  description
    "Path State Block (PSB) container.";
  list psb {
    description
      "List of Path State Blocks.";
    leaf source-port {
      type inet:port-number;
      description
        "RSVP source port.";
      reference
        "RFC2205";
    }
    leaf expires-in {
      type uint32;
      units "seconds";
      description
        "Time to expiry (in seconds).";
    }
  }
}
container rsbs {
  description
    "Reservation State Block (RSB) container.";
  list rsb {
    description
      "List of Reservation State Blocks.";
    leaf source-port {
      type inet:port-number;
      description
        "RSVP source port.";
      reference
        "RFC2205";
    }
    leaf reservation-style {
      type identityref {
        base reservation-style;
      }
      description
        "RSVP reservation style.";
    }
  }
}
```



```
    }
    leaf expires-in {
      type uint32;
      units "seconds";
      description
        "Time to expiry (in seconds).";
    }
  }
}

grouping neighbor-attributes {
  description
    "Top level grouping for RSVP neighbor properties.";
  leaf address {
    type inet:ip-address;
    description
      "Address of the RSVP neighbor.";
  }
  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 neighbor-restart-time {
    type uint32;
    units "seconds";
    description
      "Neighbor restart time.";
  }
  leaf neighbor-recovery-time {
    type uint32;
    units "seconds";
    description
      "Neighbor recovery time.";
  }
  container helper-mode {
    description
      "Helper mode information.";
    leaf enabled {
      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 neighbor-restart-time-remaining {
    type uint32;
    units "seconds";
    description
      "Number of seconds remaining for neighbor to send Hello
      message after restart.";
  }
  leaf neighbor-recovery-time-remaining {
    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
                "RSVP Hellos enabled.";
        }
        enum disabled {
            description
                "RSVP Hellos disabled.";
        }
        enum restarting {
            description
                "RSVP restarting.";
        }
    }
}
description
    "RSVP Hello status.";
}
leaf interface {
    type if:interface-ref;
    description
        "Interface where RSVP neighbor was detected.";
}
leaf neighbor-status {
    type enumeration {
        enum up {
            description
                "Neighbor state up.";
        }
        enum down {
            description
                "Neighbor state down.";
        }
        enum hello-disable {
            description
                "RSVP Hellos disabled.";
        }
        enum restarting {
            description
                "RSVP neighbor restarting.";
        }
    }
}
```



```
    }
    description
      "RSVP neighbor state.";
  }
  leaf refresh-reduction-capable {
    type boolean;
    description
      "Enables all RSVP refresh reduction message bundling, RSVP
      message ID, reliable message delivery and summary refresh.";
    reference
      "RFC2961 RSVP Refresh Overhead Reduction Extensions";
  }
  leaf restart-count {
    type yang:counter32;
    description
      "Number of times this RSVP neighbor has restarted.";
  }
  leaf restart-time {
    type yang:date-and-time;
    description
      "Last restart time of the RSVP neighbor.";
  }
}

grouping packet-statistics {
  description
    "Packet statistics grouping.";
  container packets {
    description
      "Packet statistics container.";
    leaf sent {
      type yang:counter64;
      description
        "RSVP packet sent count.";
    }
    leaf received {
      type yang:counter64;
      description
        "RSVP packet received count.";
    }
  }
}

grouping message-statistics {
  description
    "RSVP protocol statistics grouping.";
  container messages {
    description
```



```
    "RSVP protocol statistics container.";
  leaf ack-sent {
    type yang:counter64;
    description
      "RSVP Hello sent count.";
  }
  leaf ack-received {
    type yang:counter64;
    description
      "RSVP Hello received count.";
  }
  leaf bundle-sent {
    type yang:counter64;
    description
      "RSVP Bundle message sent count.";
  }
  leaf bundle-received {
    type yang:counter64;
    description
      "RSVP Bundle message received count.";
  }
  leaf hello-sent {
    type yang:counter64;
    description
      "RSVP Hello message sent count.";
  }
  leaf hello-received {
    type yang:counter64;
    description
      "RSVP Hello message received count.";
  }
  leaf integrity-challenge-sent {
    type yang:counter64;
    description
      "RSVP Integrity Challenge message sent count.";
  }
  leaf integrity-challenge-received {
    type yang:counter64;
    description
      "RSVP Integrity Challenge message received count.";
  }
  leaf integrity-response-sent {
    type yang:counter64;
    description
      "RSVP Integrity Response message sent count.";
  }
  leaf integrity-response-received {
    type yang:counter64;
```



```
    description
      "RSVP Integrity Response message received count.";
  }
  leaf notify-sent {
    type yang:counter64;
    description
      "RSVP Notify message sent count.";
  }
  leaf notify-received {
    type yang:counter64;
    description
      "RSVP Notify message received count.";
  }
  leaf path-sent {
    type yang:counter64;
    description
      "RSVP Path message sent count.";
  }
  leaf path-received {
    type yang:counter64;
    description
      "RSVP Path message received count.";
  }
  leaf path-err-sent {
    type yang:counter64;
    description
      "RSVP Path error message sent count.";
  }
  leaf path-err-received {
    type yang:counter64;
    description
      "RSVP Path error message received count.";
  }
  leaf path-tear-sent {
    type yang:counter64;
    description
      "RSVP Path tear message sent count.";
  }
  leaf path-tear-received {
    type yang:counter64;
    description
      "RSVP Path tear message received count.";
  }
  leaf resv-sent {
    type yang:counter64;
    description
      "RSVP Resv message sent count.";
  }
}
```



```
leaf resv-received {
  type yang:counter64;
  description
    "RSVP Resv message received count.";
}
leaf resv-confirm-sent {
  type yang:counter64;
  description
    "RSVP Confirm message sent count.";
}
leaf resv-confirm-received {
  type yang:counter64;
  description
    "RSVP Confirm message received count.";
}
leaf resv-err-sent {
  type yang:counter64;
  description
    "RSVP Resv error message sent count.";
}
leaf resv-err-received {
  type yang:counter64;
  description
    "RSVP Resv error message received count.";
}
leaf resv-tear-sent {
  type yang:counter64;
  description
    "RSVP Resv tear message sent count.";
}
leaf resv-tear-received {
  type yang:counter64;
  description
    "RSVP Resv tear message received count.";
}
leaf summary-refresh-sent {
  type yang:counter64;
  description
    "RSVP Summary refresh message sent count.";
}
leaf summary-refresh-received {
  type yang:counter64;
  description
    "RSVP Summary refresh message received count.";
}
leaf unknown-messages-received {
  type yang:counter64;
  description
```



```
        "Unknown messages received count.";
    }
}

grouping errors-statistics {
    description
        "Error statistics grouping.";
    container errors {
        description
            "Error statistics container.";
        leaf authenticate {
            type yang:counter64;
            description
                "The total number of RSVP packets received with an
                authentication failure.";
        }
        leaf checksum {
            type yang:counter64;
            description
                "The total number of RSVP packets received with an invalid
                checksum value.";
        }
        leaf packet-length {
            type yang:counter64;
            description
                "The total number of packets received with an invalid
                packet length.";
        }
    }
}

grouping statistics {
    description
        "RSVP statistic attributes.";
    container statistics {
        config false;
        description
            "RSVP statistics container.";
        uses message-statistics;
        uses packet-statistics;
        uses errors-statistics;
    }
}

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



```
    container sessions {
      description
        "RSVP sessions container.";
      list session-ip {
        key "destination protocol-id destination-port";
        config false;
        description
          "List of RSVP sessions.";
        uses session-attributes;
      }
    }
    uses statistics;
  }

  grouping intf-attributes {
    description
      "Top level grouping for RSVP interface properties.";
    uses signaling-parameters;
    uses refresh-reduction;
    uses hellos;
    uses authentication;
    uses statistics;
  }

  augment "/rt:routing/rt:control-plane-protocols/"
    + "rt:control-plane-protocol" {
    when "rt:type = 'rsvp:rsvp'" {
      description
        "This augment is only valid when routing protocol instance
        type is RSVP.";
    }
    description
      "RSVP protocol augmentation.";
    container rsvp {
      presence "Enable RSVP feature";
      description
        "RSVP feature container";
      container globals {
        description
          "RSVP global properties.";
        uses global-attributes;
        uses graceful-restart;
      }
      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 neighbors {
    description
        "RSVP neighbors container";
    list neighbor {
        key "address";
        description
            "List of RSVP neighbors";
        uses neighbor-attributes;
    }
}
}
}

grouping session-ref {
    description
        "Session reference information";
    leaf destination {
        type leafref {
            path "/rt:routing/rt:control-plane-protocols"
                + "/rt:control-plane-protocol/rsvp:rsvp/rsvp:globals"
                + "/rsvp:sessions/rsvp:session-ip/destination";
        }
        mandatory true;
        description
            "The RSVP session destination.";
    }
    leaf protocol-id {
        type uint8;
        mandatory true;
        description
            "The RSVP session protocol ID.";
    }
    leaf destination-port {
        type inet:ip-address;
        mandatory true;
        description
            "The RSVP session destination port.";
```



```
    }
  }

  rpc clear-session {
    nacm:default-deny-all;
    description
      "Clears RSVP sessions RPC";
    input {
      leaf routing-protocol-instance-name {
        type leafref {
          path "/rt:routing/rt:control-plane-protocols/"
            + "rt:control-plane-protocol/rt:name";
        }
        mandatory true;
        description
          "Name of the RSVP protocol instance whose session
           is being cleared.

          If the corresponding RSVP instance doesn't exist,
          then the operation will fail with an error-tag of
          'data-missing' and an error-app-tag of
          'routing-protocol-instance-not-found'.";
      }
    }
    choice filter-type {
      mandatory true;
      description
        "Filter choice";
      case match-all {
        leaf all {
          type empty;
          mandatory true;
          description
            "Match all RSVP sessions.";
        }
      }
      case match-one {
        container session-info {
          description
            "Specifies the specific session to invoke operation on";
          choice session-type {
            mandatory true;
            description
              "The RSVP session type.";
            case rsvp-session-ip {
              uses session-ref;
            }
          }
        }
      }
    }
  }
}
```



```
    }
  }
}

rpc clear-neighbor {
  nacm:default-deny-all;
  description
    "RPC to clear the RSVP Hello session to a neighbor.";
  input {
    leaf routing-protocol-instance-name {
      type leafref {
        path "/rt:routing/rt:control-plane-protocols/"
          + "rt:control-plane-protocol/rt:name";
      }
      mandatory true;
      description
        "Name of the RSVP protocol instance whose session
        is being cleared.

        If the corresponding RSVP instance doesn't exist,
        then the operation will fail with an error-tag of
        'data-missing' and an error-app-tag of
        'routing-protocol-instance-not-found'.";
    }
  }
  choice filter-type {
    mandatory true;
    description
      "The Filter choice.";
    case match-all {
      leaf all {
        type empty;
        mandatory true;
        description
          "Match all RSVP neighbor sessions.";
      }
    }
    case match-one {
      leaf neighbor-address {
        type leafref {
          path "/rt:routing/rt:control-plane-protocols"
            + "/rt:control-plane-protocol/rsvp:rsvp"
            + "/rsvp:neighbors/rsvp:neighbor/address";
        }
        mandatory true;
        description
          "Match the specific RSVP neighbor session.";
      }
    }
  }
}
```



```
    }
  }
}

rpc clear-authentication {
  nacm:default-deny-all;
  description
    "Clears the RSVP Security Association (SA) before the
    lifetime expires.";
  input {
    leaf routing-protocol-instance-name {
      type leafref {
        path "/rt:routing/rt:control-plane-protocols/"
          + "rt:control-plane-protocol/rt:name";
      }
      mandatory true;
      description
        "Name of the RSVP protocol instance whose session
        is being cleared.

        If the corresponding RSVP instance doesn't exist,
        then the operation will fail with an error-tag of
        'data-missing' and an error-app-tag of
        'routing-protocol-instance-not-found'.";
    }
    choice filter-type {
      mandatory true;
      description
        "Filter choice";
      case match-all {
        leaf all {
          type empty;
          mandatory true;
          description
            "Match all RSVP security associations.";
        }
      }
      case match-one-interface {
        leaf interface {
          type if:interface-ref;
          description
            "Interface where RSVP security association(s) to be
            detected.";
        }
      }
    }
  }
}
```



```

    }
  }
<CODE ENDS>

```

## 5. RSVP Extended YANG Model

The RSVP extended module augments the RSVP base module with additional and optional feature data.

### 5.1. Module Structure

The RSVP extended YANG module covers non-core RSVP feature(s). It also covers feature(s) that MAY be supported vendors claiming support for RSVP protocol.

### 5.2. Tree Diagram

Figure 4 shows the YANG tree representation for configuration and state data that is augmenting the RSVP extended module:

```

module: ietf-rsvp-extended
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:globals
    /rsvp:graceful-restart:
      +--rw restart-time?      uint32
      +--rw recovery-time?    uint32
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:globals
    /rsvp:statistics/rsvp:packets:
      +--ro discontinuity-time?  yang:date-and-time
      +--ro out-dropped?        yang:counter64
      +--ro in-dropped?         yang:counter64
      +--ro out-errors?         yang:counter64
      +--ro in-errors?          yang:counter64
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:globals
    /rsvp:statistics/rsvp:messages:
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:globals
    /rsvp:statistics/rsvp:errors:
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces:
      +--rw refresh-interval?    uint32
      +--rw refresh-misses?     uint32
      +--rw checksum?           boolean
      +--rw patherr-state-removal? empty
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces

```



```
        /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 /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces
    /rsvp:hellos:
+--rw interface-based?    empty
+--rw hello-interval?     uint32
+--rw hello-misses?       uint32
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces
    /rsvp:authentication:
+--rw lifetime?           uint32
+--rw window-size?        uint32
+--rw challenge?          empty
+--rw retransmits?        uint32
+--rw key-chain?          key-chain:key-chain-ref
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces
    /rsvp:interface:
+--rw refresh-interval?    uint32
+--rw refresh-misses?     uint32
+--rw checksum?           boolean
+--rw patherr-state-removal? empty
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces
    /rsvp:interface/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 /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces
    /rsvp:interface/rsvp:hellos:
+--rw interface-based?    empty
+--rw hello-interval?     uint32
+--rw hello-misses?       uint32
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces
    /rsvp:interface/rsvp:authentication:
+--rw lifetime?           uint32
+--rw window-size?        uint32
```



```
+--rw challenge?      empty
+--rw retransmits?    uint32
+--rw key-chain?      key-chain:key-chain-ref
```

Figure 4: RSVP extended module tree diagram

### 5.3. YANG Module

The `ietf-rsvp-extended` module imports from the following modules:

- o `ietf-rsvp` defined in this document
- o `ietf-routing` defined in [[RFC8349](#)]
- o `ietf-yang-types` and `ietf-inet-types` defined in [[RFC6991](#)]
- o `ietf-key-chain` defined in [[RFC8177](#)]

Figure 5 shows the RSVP extended YANG module:

This module references the following documents: [[RFC2747](#)], [[RFC3209](#)], and [[RFC5495](#)].

```
<CODE BEGINS> file "ietf-rsvp-extended@2020-07-24.yang"
module ietf-rsvp-extended {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-rsvp-extended";
  prefix rsvp-ext;

  import ietf-rsvp {
    prefix rsvp;
    reference
      "RFCXXXX: A YANG Data Model for Resource Reservation Protocol
      (RSVP)";
  }
  import ietf-routing {
    prefix rt;
    reference
      "RFC8349: A YANG Data Model for Routing Management
      (NMDA Version)";
  }
  import ietf-yang-types {
    prefix yang;
    reference
      "RFC6991: Common YANG Data Types";
  }
  import ietf-key-chain {
    prefix key-chain;
```



```
reference
  "RFC8177: YANG Data Model for Key Chains";
}

organization
  "IETF Traffic Engineering Architecture and Signaling (TEAS)
  Working Group";
contact
  "WG Web:  <http://tools.ietf.org/wg/teas/>
  WG List:  <mailto:teas@ietf.org>

  Editor:    Vishnu Pavan Beeram
             <mailto:vbeeram@juniper.net>

  Editor:    Tarek Saad
             <mailto:tsaad@juniper.net>

  Editor:    Rakesh Gandhi
             <mailto:rgandhi@cisco.com>

  Editor:    Xufeng Liu
             <mailto:xufeng.liu.ietf@gmail.com>

  Editor:    Igor Bryskin
             <mailto:i\_bryskin@yahoo.com>";

description
  "This module contains the Extended RSVP YANG data model.
  The model fully conforms to the Network Management Datastore
  Architecture (NMDA).

  Copyright (c) 2019 IETF Trust and the persons
  identified as authors of the code.  All rights reserved.

  Redistribution and use in source and binary forms, with or
  without modification, is permitted pursuant to, and subject
  to the license terms contained in, the Simplified BSD License
  set forth in Section 4.c of the IETF Trust's Legal Provisions
  Relating to IETF Documents
  (https://trustee.ietf.org/license-info).
  This version of this YANG module is part of RFC XXXX; see
  the RFC itself for full legal notices.";

// RFC Ed.: replace XXXX with actual RFC number and remove this
// note.
// RFC Ed.: update the date below with the date of RFC publication
// and remove this note.

revision 2020-07-24 {
```



```
    description
      "Initial version.";
    reference
      "RFCXXXX: A YANG Data Model for Resource Reservation Protocol
        (RSVP)";
  }

  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
        "RFC5495: Description of the Resource Reservation Protocol -
          Traffic-Engineered (RSVP-TE) Graceful Restart Procedures.";
    }
    leaf recovery-time {
      type uint32;
      units "seconds";
      description
        "RSVP state recovery time.";
    }
  }

  grouping authentication-extended-config {
    description
      "Configuration parameters relating to RSVP authentication.";
    leaf lifetime {
      type uint32 {
        range "30..86400";
      }
      units "seconds";
      description
        "Life time for each security association.";
      reference
        "RFC2747: RSVP Cryptographic Authentication";
    }
    leaf window-size {
      type uint32 {
        range "1..64";
      }
      description
        "Window-size to limit number of out-of-order messages.";
      reference
        "RFC2747: RSVP Cryptographic Authentication";
    }
  }
}
```



```
}
leaf challenge {
  type empty;
  description
    "Enable challenge messages.";
  reference
    "RFC2747: RSVP Cryptographic Authentication";
}
leaf retransmits {
  type uint32 {
    range "1..10000";
  }
  description
    "Number of retransmits when messages are
    dropped.";
  reference
    "RFC2747: RSVP Cryptographic
    Authentication";
}
leaf key-chain {
  type key-chain:key-chain-ref;
  description
    "Key chain name to authenticate RSVP
    signaling messages.";
  reference
    "RFC2747: RSVP Cryptographic Authentication";
}
}

grouping 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
      "RFC3209: RSVP-TE: Extensions to RSVP for LSP Tunnels.
      RFC5495: 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
    "RFC3209: RSVP-TE: Extensions to RSVP for LSP Tunnels.
    RFC5495: 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 boolean;
    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 packets-extended-statistics {
    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 out-dropped {
        type yang:counter64;
        description
            "Out RSVP packet drop count.";
    }
    leaf in-dropped {
        type yang:counter64;
        description
            "In RSVP packet drop count.";
    }
    leaf out-errors {
        type yang:counter64;
        description
            "Out RSVP packet errors count.";
    }
    leaf in-errors {
        type yang:counter64;
        description
            "In RSVP packet rx errors count.";
    }
}

/**
 * RSVP extensions augmentations
 */
/* RSVP globals graceful restart*/

augment "/rt:routing/rt:control-plane-protocols/"
    + "rt:control-plane-protocol/rsvp:rsvp/rsvp:globals/"
    + "rsvp:graceful-restart" {
    description
        "RSVP globals configuration extensions";
    uses graceful-restart-extended-config;
}

/* RSVP statistics augmentation */

augment "/rt:routing/rt:control-plane-protocols/"
    + "rt:control-plane-protocol/rsvp:rsvp/rsvp:globals/"
    + "rsvp:statistics/rsvp:packets" {
    description
        "RSVP packet stats extensions";
    uses packets-extended-statistics;
}
```



```
/**
 * RSVP all interfaces extensions
 */
/* RSVP interface signaling extensions */

augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces" {
  description
    "RSVP signaling all interfaces configuration extensions";
  uses signaling-parameters-extended-config;
}

/* RSVP refresh reduction extension */

augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
  + "rsvp:refresh-reduction" {
  description
    "RSVP refresh-reduction all interface configuration
    extensions";
  uses refresh-reduction-extended-config;
}

/* RSVP hellos extension */

augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
  + "rsvp:hellos" {
  description
    "RSVP hello all interfaces configuration extensions";
  uses hellos-extended-config;
}

/* RSVP authentication extension */

augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
  + "rsvp:authentication" {
  description
    "RSVP authentication all interfaces configuration extensions";
  uses authentication-extended-config;
}

/**
 * RSVP interface extensions
 */
/* RSVP interface signaling extensions */
```



```
augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
  + "rsvp:interface" {
  description
    "RSVP signaling interface configuration extensions";
  uses signaling-parameters-extended-config;
}

/* RSVP refresh reduction extension */

augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
  + "rsvp:interface/rsvp:refresh-reduction" {
  description
    "RSVP refresh-reduction interface configuration extensions";
  uses refresh-reduction-extended-config;
}

/* RSVP hellos extension */

augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
  + "rsvp:interface/rsvp:hellos" {
  description
    "RSVP hello interface configuration extensions";
  uses hellos-extended-config;
}

/* RSVP authentication extension */

augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
  + "rsvp:interface/rsvp:authentication" {
  description
    "RSVP authentication interface configuration extensions";
  uses authentication-extended-config;
}
}
<CODE ENDS>
```

Figure 5: RSVP extended YANG module

## 6. 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  
Registrant Contact: The IESG.  
XML: N/A, the requested URI is an XML namespace.

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

This document registers two YANG modules in the YANG Module Names registry [[RFC6020](#)].

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

name: iETF-rsvp-extended  
namespace: urn:ietf:params:xml:ns:yang:ietf-rsvp-extended  
prefix: rsvp-extended  
reference: RFCXXXX

## 7. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [[RFC6241](#)] or RESTCONF [[RFC8040](#)]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [[RFC6242](#)]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [[RFC8446](#)].

The Network Configuration Access Control Model (NACM) [[RFC8341](#)] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in the YANG module(s) defined in this document that 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. These are the subtrees and data nodes and their sensitivity/vulnerability:

/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/  
rsvp:rsvp/ /rsvp:globals /rsvp:interfaces /rsvp:sessions



All of which are considered sensitive and if access to either of these is compromised, it can result in temporary network outages or be employed to mount DoS attacks.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

```
/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/  
rsvp:rsvp/ /rsvp:globals /rsvp:interfaces /rsvp:sessions
```

Additional information from these state data nodes can be inferred with respect to the network topology, and device location and subsequently be used to mount other attacks in the network.

For RSVP authentication, the configuration supported is via the specification of key-chains [[RFC8177](#)] or the direct specification of key and authentication algorithm, and hence security considerations of [[RFC8177](#)] are inherited. This includes the considerations with respect to the local storage and handling of authentication keys.

Some of the RPC operations defined in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations. The RSVP YANG module support the "clear-session" and "clear-neighbor" RPCs. If access to either of these is compromised, they can result in temporary network outages be employed to mount DoS attacks.

The security considerations spelled out in the YANG 1.1 specification [[RFC7950](#)] apply for this document as well.

## **8. Acknowledgement**

The authors would like to thank Tom Petch for reviewing and providing useful feedback about the document. The authors would also like to thank Lou Berger for reviewing and providing valuable feedback on this document.

## **9. Contributors**



Himanshu Shah  
Ciena

Email: hshah@ciena.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

## **10. References**

### **10.1. Normative References**

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", [BCP 81](#), [RFC 3688](#), DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", [RFC 6020](#), DOI 10.17487/RFC6020, October 2010, <<https://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, <<https://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, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", [RFC 6991](#), DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", [RFC 7950](#), DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", [RFC 8040](#), DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in [RFC 2119](#) Key Words", [BCP 14](#), [RFC 8174](#), DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8177] Lindem, A., Ed., Qu, Y., Yeung, D., Chen, I., and J. Zhang, "YANG Data Model for Key Chains", [RFC 8177](#), DOI 10.17487/RFC8177, June 2017, <<https://www.rfc-editor.org/info/rfc8177>>.
- [RFC8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Common YANG Data Types for the Routing Area", [RFC 8294](#), DOI 10.17487/RFC8294, December 2017, <<https://www.rfc-editor.org/info/rfc8294>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", [BCP 215](#), [RFC 8340](#), DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, [RFC 8341](#), DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", [RFC 8343](#), DOI 10.17487/RFC8343, March 2018, <<https://www.rfc-editor.org/info/rfc8343>>.
- [RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", [RFC 8349](#), DOI 10.17487/RFC8349, March 2018, <<https://www.rfc-editor.org/info/rfc8349>>.



- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", [RFC 8446](#), DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8639] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Subscription to YANG Notifications", [RFC 8639](#), DOI 10.17487/RFC8639, September 2019, <<https://www.rfc-editor.org/info/rfc8639>>.
- [RFC8641] Clemm, A. and E. Voit, "Subscription to YANG Notifications for Datastore Updates", [RFC 8641](#), DOI 10.17487/RFC8641, September 2019, <<https://www.rfc-editor.org/info/rfc8641>>.

## **10.2. Informative References**

- [I-D.ietf-teas-yang-rsvp-te]  
Beeram, V., Saad, T., Gandhi, R., Liu, X., Bryskin, I., and H. Shah, "A YANG Data Model for RSVP-TE Protocol", [draft-ietf-teas-yang-rsvp-te-08](#) (work in progress), March 2020.
- [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, <<https://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, <<https://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, <<https://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, <<https://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, <<https://www.rfc-editor.org/info/rfc3473>>.



- [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, <<https://www.rfc-editor.org/info/rfc5063>>.
- [RFC5495] Li, D., Gao, J., Satyanarayana, A., and S. Bardalai, "Description of the Resource Reservation Protocol - Traffic-Engineered (RSVP-TE) Graceful Restart Procedures", [RFC 5495](#), DOI 10.17487/RFC5495, March 2009, <<https://www.rfc-editor.org/info/rfc5495>>.

#### Authors' Addresses

Vishnu Pavan Beeram  
Juniper Networks

Email: [vbeeram@juniper.net](mailto:vbeeram@juniper.net)

Tarek Saad  
Juniper Networks

Email: [tsaad@juniper.net](mailto:tsaad@juniper.net)

Rakesh Gandhi  
Cisco Systems, Inc.

Email: [rgandhi@cisco.com](mailto:rgandhi@cisco.com)

Xufeng Liu  
Volta Networks

Email: [xufeng.liu.ietf@gmail.com](mailto:xufeng.liu.ietf@gmail.com)

Igor Bryskin  
Individual

Email: [i\\_bryskin@yahoo.com](mailto:i_bryskin@yahoo.com)

