

Internet Engineering Task Force (IETF)  
Request for Comments: 9130  
Category: Standards Track  
ISSN: 2070-1721

S. Litkowski, Ed.  
Cisco Systems  
D. Yeung  
Arrcus, Inc.  
A. Lindem  
Cisco Systems  
J. Zhang  
Juniper Networks  
L. Lhotka  
CZ.NIC  
October 2022

## YANG Data Model for the IS-IS Protocol

### Abstract

This document defines a YANG data model that can be used to configure and manage the IS-IS protocol on network elements.

### Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <https://www.rfc-editor.org/info/rfc9130>.

### Copyright Notice

Copyright (c) 2022 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

### Table of Contents

1. Introduction
  - 1.1. Requirements Language
2. Design of the Data Model
  - 2.1. IS-IS Configuration
  - 2.2. Multi-Topology Parameters
  - 2.3. Per-Level Parameters
  - 2.4. Per-Interface Parameters
  - 2.5. Authentication Parameters
  - 2.6. IGP/LDP Synchronization
  - 2.7. ISO Parameters
  - 2.8. IP FRR

2.9.	Operational States
3.	RPC Operations
4.	Notifications
5.	Interactions with Other YANG Modules
6.	IS-IS YANG Module
7.	Security Considerations
8.	IANA Considerations
9.	References
9.1.	Normative References
9.2.	Informative References
Appendix A.	Example of IS-IS Configuration in XML
Acknowledgments	
Contributors	
Authors' Addresses	

## 1. Introduction

This document defines a YANG data model [RFC7950] for the IS-IS routing protocol.

The data model covers the configuration of an IS-IS routing protocol instance, as well as the retrieval of IS-IS operational states.

A simplified tree representation of the data model is presented in Section 2. Tree diagrams used in this document follow the notation defined in [RFC8340].

The module defined in this document is designed as per the Network Management Datastore Architecture (NMDA) [RFC8342].

### 1.1. 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.

## 2. Design of the Data Model

The IS-IS YANG module augments the "control-plane-protocol" list in the "ietf-routing" module [RFC8349] with specific IS-IS parameters.

The diagram below shows the overall structure of the "ietf-isis" YANG module defined in this document. NOTE: '\' line wrapping is per [RFC8792].

```

module: ietf-isis
  augment /rt:routing/rt:ribs/rt:rib/rt:routes/rt:route:
    +--ro metric?          uint32
    +--ro tag*             uint64
    +--ro route-type?     enumeration
  augment /if:interfaces/if:interface:
    +--rw clns-mtu?      uint16 {osi-interface}?
  augment /rt:routing/rt:control-plane-protocols/rt:
    control-plane-protocol:
      +--rw isis
        +--rw enabled?          boolean {admin-control}?
        +--rw level-type?       level
        +--rw system-id?        system-id
        +--rw maximum-area-addresses? \
          uint8 {maximum-area-addresses}?
        +--rw area-address*      area-address
        +--rw lsp-mtu?           uint16
        +--rw lsp-lifetime?      uint16
        +--rw lsp-refresh?       rt-types:timer-value-seconds16

```

```

|                                     {lsp-refresh}?
+--rw poi-tlv?                       boolean {poi-tlv}?
+--rw graceful-restart {graceful-restart}?
|   +--rw enabled?                   boolean
|   +--rw restart-interval?          rt-types:timer-value-seconds16
|   +--rw helper-enabled?            boolean
+--rw nsr {nsr}?
|   +--rw enabled?                   boolean
+--rw node-tags {node-tag}?
|   +--rw node-tag* [tag]
|   ...
+--rw metric-type
|   +--rw value?                     enumeration
|   +--rw level-1
|   |   ...
|   +--rw level-2
|   ...
+--rw default-metric
|   +--rw value?                     wide-metric
|   +--rw level-1
|   |   ...
|   +--rw level-2
|   ...
+--rw auto-cost {auto-cost}?
|   +--rw enabled?                   boolean
|   +--rw reference-bandwidth?       uint32
+--rw authentication
|   +--rw (authentication-type)?
|   |   ...
|   +--rw level-1
|   |   ...
|   +--rw level-2
|   ...
+--rw address-families {nlpid-control}?
|   +--rw address-family-list* [address-family]
|   ...
+--rw mpls
|   +--rw te-rid {te-rid}?
|   |   ...
|   +--rw ldp
|   ...
+--rw spf-control
|   +--rw paths?                     uint16 {max-ecmp}?
|   +--rw ietf-spf-delay {ietf-spf-delay}?
|   ...
+--rw fast-reroute {fast-reroute}?
|   +--rw lfa {lfa}?
+--rw preference
|   +--rw (granularity)?
|   ...
+--rw overload
|   +--rw status?                   boolean
+--rw overload-max-metric {overload-max-metric}?
|   +--rw timeout?                  rt-types:timer-value-seconds16
+--ro spf-log
|   +--ro event* [id]
|   ...
+--ro lsp-log
|   +--ro event* [id]
|   ...
+--ro hostnames
|   +--ro hostname* [system-id]
|   ...
+--ro database
|   +--ro levels* [level]
|   ...

```

```

+--ro local-rib
|   +--ro route* [prefix]
|   ...
+--ro system-counters
|   +--ro level* [level]
|   ...
+--ro protected-routes
|   +--ro address-family-stats* \
|       [address-family prefix alternate]
|   ...
+--ro unprotected-routes
|   +--ro prefixes* [address-family prefix]
|   ...
+--ro protection-statistics* [frr-protection-method]
|   +--ro frr-protection-method    identityref
|   +--ro address-family-stats* [address-family]
|   ...
+--rw discontinuity-time?          yang:date-and-time
+--rw topologies {multi-topology}?
|   +--rw topology* [name]
|   ...
+--rw interfaces
|   +--rw interface* [name]
|   ...

```

#### rpcs:

```

+---x clear-adjacency
|   +---w input
|       +---w routing-protocol-instance-name -> /rt:routing/
|                                               control-plane-\
|                                               protocols/
|                                               control-plane-\
|                                               protocol/name
|   +---w level?                             level
|   +---w interface?                         if:interface-ref
+---x clear-database
|   +---w input
|       +---w routing-protocol-instance-name -> /rt:routing/
|                                               control-plane-\
|                                               protocols/
|                                               control-plane-\
|                                               protocol/name
|   +---w level?                             level

```

#### notifications:

```

+---n database-overload
|   +--ro routing-protocol-name? -> /rt:routing/
|                                   control-plane-protocols/
|                                   control-plane-protocol/name
|   +--ro isis-level?            level
|   +--ro overload?              enumeration
+---n lsp-too-large
|   +--ro routing-protocol-name? -> /rt:routing/
|                                   control-plane-protocols/
|                                   control-plane-protocol/name
|   +--ro isis-level?            level
|   +--ro interface-name?        if:interface-ref
|   +--ro interface-level?       level
|   +--ro extended-circuit-id?   extended-circuit-id
|   +--ro pdu-size?              uint32
|   +--ro lsp-id?                lsp-id
+---n if-state-change
|   +--ro routing-protocol-name? -> /rt:routing/
|                                   control-plane-protocols/
|                                   control-plane-protocol/name
|   +--ro isis-level?            level

```



```

control-plane-protocol/name
+--ro isis-level?          level
+--ro interface-name?      if:interface-ref
+--ro interface-level?     level
+--ro extended-circuit-id? extended-circuit-id
+--ro raw-pdu?             binary
+---n version-skew
+--ro routing-protocol-name? -> /rt:routing/
|                               control-plane-protocols/
|                               control-plane-protocol/name
+--ro isis-level?          level
+--ro interface-name?      if:interface-ref
+--ro interface-level?     level
+--ro extended-circuit-id? extended-circuit-id
+--ro protocol-version?    uint8
+--ro raw-pdu?             binary
+---n area-mismatch
+--ro routing-protocol-name? -> /rt:routing/
|                               control-plane-protocols/
|                               control-plane-protocol/name
+--ro isis-level?          level
+--ro interface-name?      if:interface-ref
+--ro interface-level?     level
+--ro extended-circuit-id? extended-circuit-id
+--ro raw-pdu?             binary
+---n rejected-adjacency
+--ro routing-protocol-name? -> /rt:routing/
|                               control-plane-protocols/
|                               control-plane-protocol/name
+--ro isis-level?          level
+--ro interface-name?      if:interface-ref
+--ro interface-level?     level
+--ro extended-circuit-id? extended-circuit-id
+--ro raw-pdu?             binary
+--ro reason?              string
+---n protocols-supported-mismatch
+--ro routing-protocol-name? -> /rt:routing/
|                               control-plane-protocols/
|                               control-plane-protocol/name
+--ro isis-level?          level
+--ro interface-name?      if:interface-ref
+--ro interface-level?     level
+--ro extended-circuit-id? extended-circuit-id
+--ro raw-pdu?             binary
+--ro protocols*           uint8
+---n lsp-error-detected
+--ro routing-protocol-name? -> /rt:routing/
|                               control-plane-protocols/
|                               control-plane-protocol/name
+--ro isis-level?          level
+--ro interface-name?      if:interface-ref
+--ro interface-level?     level
+--ro extended-circuit-id? extended-circuit-id
+--ro lsp-id?              lsp-id
+--ro raw-pdu?             binary
+--ro error-offset?        uint32
+--ro tlv-type?            uint8
+---n adjacency-state-change
+--ro routing-protocol-name? -> /rt:routing/
|                               control-plane-protocols/
|                               control-plane-protocol/name
+--ro isis-level?          level
+--ro interface-name?      if:interface-ref
+--ro interface-level?     level
+--ro extended-circuit-id? extended-circuit-id
+--ro neighbor?            string

```

```

|   +---ro neighbor-system-id?      system-id
|   +---ro state?                   adj-state-type
|   +---ro reason?                   string
+---n lsp-received
|   +---ro routing-protocol-name?   -> /rt:routing/
|   |                               control-plane-protocols/
|   |                               control-plane-protocol/name
|   +---ro isis-level?              level
|   +---ro interface-name?          if:interface-ref
|   +---ro interface-level?         level
|   +---ro extended-circuit-id?     extended-circuit-id
|   +---ro lsp-id?                  lsp-id
|   +---ro sequence?                uint32
|   +---ro received-timestamp?      yang:timestamp
|   +---ro neighbor-system-id?      system-id
+---n lsp-generation
|   +---ro routing-protocol-name?   -> /rt:routing/
|   |                               control-plane-protocols/
|   |                               control-plane-protocol/name
|   +---ro isis-level?              level
|   +---ro lsp-id?                  lsp-id
|   +---ro sequence?                uint32
|   +---ro send-timestamp?          yang:timestamp

```

## 2.1. IS-IS Configuration

The IS-IS configuration is divided into two areas:

- \* Global parameters
- \* Per-interface configuration (see Section 2.4)

Additional modules may be created to support additional parameters. These additional modules **MUST** augment the "ietf-isis" module.

The model includes optional features for which the corresponding configuration data nodes are also optional. As an example, the ability to control the administrative state of a particular IS-IS instance is optional. By advertising the feature "admin-control", a device communicates to the client that it supports the ability to shut down a particular IS-IS instance.

The global configuration contains the usual IS-IS parameters, e.g., "lsp-mtu", "lsp-lifetime", "lsp-refresh", "default-metric".

## 2.2. Multi-Topology Parameters

The model supports Multi-Topology (MT) IS-IS as defined in [RFC5120].

The "topologies" container is used to enable support of the MT extensions.

The "name" used in the topology list should refer to an existing Routing Information Base (RIB) defined for the device [RFC8349].

Some specific parameters can be defined on a per-topology basis, at both the global level and the interface level: for example, an interface metric can be defined per topology.

Multiple address families (such as IPv4 or IPv6) can also be enabled within the default topology. This can be achieved using the "address-families" container (requiring the "nlpid-control" feature to be supported).

## 2.3. Per-Level Parameters

Some parameters allow per-level configuration. For such parameters, the parameter is modeled as a container with three configuration locations:

A top-level container:

Corresponds to "level-all", so the configuration applies to both levels.

A level-1 container:

Corresponds to parameters specific to level 1.

A level-2 container:

Corresponds to parameters specific to level 2.

```

+--rw priority
|   +--rw value?      uint8
|   +--rw level-1
|       |   +--rw value?      uint8
|       +--rw level-2
|           +--rw value?      uint8

```

Example:

```

<priority>
  <value>250</value>
  <level-1>
    <value>100</value>
  </level-1>
</priority>

```

An implementation **MUST** prefer a level-specific parameter over a top-level parameter. For example, if the priority is 100 for the level-1 configuration and 250 for the top-level configuration, the implementation must use 100 for the level-1 priority and 250 for the level-2 priority.

Some parameters, such as "overload bit" and "route preference", are not modeled to support per-level configuration. If an implementation supports per-level configuration for such parameters, this implementation **MUST** augment the current model by adding both level-1 and level-2 containers and **MUST** reuse existing configuration groupings.

Example of augmentation:

```

augment "/rt:routing/" +
  "rt:control-plane-protocols/rt:control-plane-protocol"+
  "/isis:isis/isis:overload" {
  when "rt:type = 'isis:isis'" {
    description
      "This augments the IS-IS routing protocol when used.";
  }
  description
    "This augments the IS-IS overload configuration
    with per-level configuration.";

  container level-1 {
    uses isis:overload-global-cfg;
    description
      "Level-1 configuration.";
  }
  container level-2 {
    uses isis:overload-global-cfg;
    description
      "Level-2 configuration.";
  }
}

```

```
}
```

If an implementation does not support per-level configuration for a parameter modeled with per-level configuration, the implementation should advertise a deviation to announce that it does not support the configuration of level-1 and level-2 containers.

Finally, if an implementation supports per-level configuration but does not support the "level-all" configuration, it should also advertise a deviation.

## 2.4. Per-Interface Parameters

The per-interface section of the IS-IS instance describes the interface-specific parameters.

The interface is modeled as a reference to an existing interface as defined in the "ietf-interfaces" YANG module [RFC8343].

Each interface has some interface-specific parameters that may have a different per-level value as described in the previous section. An interface-specific parameter **MUST** be preferred over an IS-IS global parameter.

Some parameters, such as "hello-padding", are defined as containers to allow easy extension by vendor-specific modules.

```
+--rw interfaces
  +--rw interface* [name]
    +--rw name                               if:interface-ref
    +--rw enabled?                           boolean {admin-control}?
    +--rw level-type?                         level
    +--rw lsp-pacing-interval?                rt-types:
    |                                         timer-value-\
    |                                         milliseconds
    +--rw lsp-retransmit-interval?            rt-types:
    |                                         timer-value-seconds16
    +--rw passive?                           boolean
    +--rw csnp-interval?                      rt-types:
    |                                         timer-value-seconds16
    +--rw hello-padding
    |   +--rw enabled? boolean
    +--rw mesh-group-enabled?                mesh-group-state
    +--rw mesh-group?                        uint8
    +--rw interface-type?                    interface-type
    +--rw tag*                               uint32 {prefix-tag}?
    +--rw tag64*                             uint64 {prefix-tag64}?
    +--rw node-flag?                         boolean {node-flag}?
    +--rw hello-authentication
    |   +--rw (authentication-type)?
    |   |   +--:(key-chain) {key-chain}?
    |   |   |   +--rw key-chain?            key-chain:key-chain-ref
    |   |   +--:(password)
    |   |   |   +--rw key?                   string
    |   |   |   +--rw crypto-algorithm?     identityref
    |   +--rw level-1
    |   |   +--rw (authentication-type)?
    |   |   |   +--:(key-chain) {key-chain}?
    |   |   |   |   +--rw key-chain?        key-chain:\
    |   |   |   |   |                       key-chain-ref
    |   |   |   +--:(password)
    |   |   |   |   +--rw key?               string
    |   |   |   |   +--rw crypto-algorithm?  identityref
    |   +--rw level-2
    |   |   +--rw (authentication-type)?
    |   |   |   +--:(key-chain) {key-chain}?
```

```

|         |   +--rw key-chain?           key-chain:\
|         |   |               key-chain-ref
|         +--:(password)
|         |   +--rw key?               string
|         |   +--rw crypto-algorithm?  identityref
+--rw hello-interval
|   +--rw value?      rt-types:timer-value-seconds16
|   +--rw level-1
|   |   +--rw value?  rt-types:timer-value-seconds16
|   +--rw level-2
|   |   +--rw value?  rt-types:timer-value-seconds16
+--rw hello-multiplier
|   +--rw value?      uint16
|   +--rw level-1
|   |   +--rw value?  uint16
|   +--rw level-2
|   |   +--rw value?  uint16
+--rw priority
|   +--rw value?      uint8
|   +--rw level-1
|   |   +--rw value?  uint8
|   +--rw level-2
|   |   +--rw value?  uint8
+--rw metric
|   +--rw value?      wide-metric
|   +--rw level-1
|   |   +--rw value?  wide-metric
|   +--rw level-2
|   |   +--rw value?  wide-metric
+--rw bfd {bfd}?
|   +--rw enabled?          boolean
|   +--rw local-multiplier? multiplier
|   |   {client-base-cfg-parms}?
+--rw (interval-config-type)? {client-base-cfg-parms}?
|   +--:(tx-rx-intervals)
|   |   +--rw desired-min-tx-interval?  uint32
|   |   +--rw required-min-rx-interval?  uint32
|   +--:(single-interval) {single-minimum-interval}?
|   |   +--rw min-interval?              uint32
+--rw address-families {nlpid-control}?
|   +--rw address-family-list* [address-family]
|   +--rw address-family      iana-rt-types:address-family
+--rw mpls
|   +--rw ldp
|   |   +--rw igp-sync?  boolean {ldp-igp-sync}?
+--rw fast-reroute {fast-reroute}?
|   +--rw lfa {lfa}?
|   |   +--rw candidate-enabled?  boolean
|   |   +--rw enabled?            boolean
|   |   +--rw remote-lfa {remote-lfa}?
|   |   |   +--rw enabled?  boolean
|   +--rw level-1
|   |   +--rw candidate-enabled?  boolean
|   |   +--rw enabled?            boolean
|   |   +--rw remote-lfa {remote-lfa}?
|   |   |   +--rw enabled?  boolean
|   +--rw level-2
|   |   +--rw candidate-enabled?  boolean
|   |   +--rw enabled?            boolean
|   |   +--rw remote-lfa {remote-lfa}?
|   |   |   +--rw enabled?  boolean
+--ro adjacencies
|   +--ro adjacency* []
|   |   +--ro neighbor-sys-type?          level
|   |   +--ro neighbor-sysid?             system-id
|   |   +--ro neighbor-extended-circuit-id? extended-\

```

```

|         |
|         |         circuit-id
|         |         snpa
|         |         level
|         |         rt-types:
|         |         timer-value-\
|         |         seconds16
|         |         +---ro neighbor-priority?      uint8
|         |         +---ro lastuptime?             yang:timestamp
|         |         +---ro state?                   adj-state-type
+---ro event-counters
|   +---ro adjacency-changes?      uint32
|   +---ro adjacency-number?       uint32
|   +---ro init-fails?             uint32
|   +---ro adjacency-rejects?      uint32
|   +---ro id-len-mismatch?        uint32
|   +---ro max-area-addresses-mismatch?  uint32
|   +---ro authentication-type-fails?  uint32
|   +---ro authentication-fails?      uint32
|   +---ro lan-dis-changes?         uint32
+---ro packet-counters
|   +---ro level* [level]
|   |   +---ro level      level-number
|   |   +---ro iih
|   |   |   +---ro in?      uint32
|   |   |   +---ro out?     uint32
|   |   +---ro ish
|   |   |   +---ro in?      uint32
|   |   |   +---ro out?     uint32
|   |   +---ro esh
|   |   |   +---ro in?      uint32
|   |   |   +---ro out?     uint32
|   |   +---ro lsp
|   |   |   +---ro in?      uint32
|   |   |   +---ro out?     uint32
|   |   +---ro psnp
|   |   |   +---ro in?      uint32
|   |   |   +---ro out?     uint32
|   |   +---ro csnp
|   |   |   +---ro in?      uint32
|   |   |   +---ro out?     uint32
|   |   +---ro unknown
|   |   |   +---ro in?      uint32
+---rw discontinuity-time?          yang:date-and-time
+---rw topologies {multi-topology}?
|   +---rw topology* [name]
|   |   +---rw name      ->
|   |   |               ../../../../../../../rt:\
|   |   |               ribs/rib/name
|   |   +---rw metric
|   |   |   +---rw value?      wide-metric
|   |   |   +---rw level-1
|   |   |   |   +---rw value?      wide-metric
|   |   |   +---rw level-2
|   |   |   |   +---rw value?      wide-metric

```

## 2.5. Authentication Parameters

The module defined in this document enables authentication configuration through the "ietf-key-chain" module [RFC8177]. The IS-IS module imports the "ietf-key-chain" module and reuses some groupings to allow global and per-interface configuration of authentication. If global authentication is configured, an implementation SHOULD authenticate PSNPs (Partial Sequence Number PDUs), CSNPs (Complete Sequence Number PDUs), and LSPs (Link State PDUs) with the authentication parameters supplied. The authentication of Hello PDUs (Protocol Data Units) can be activated

on a per-interface basis.

## 2.6. IGP/LDP Synchronization

[RFC5443] defines a mechanism where IGP (the Interior Gateway Protocol) needs to be synchronized with LDP (the Label Distribution Protocol). An "ldp-igp-sync" feature has been defined in the model to support this functionality. The "mpls/ldp/igp-sync" leaf under "interface" allows activation of the functionality on a per-interface basis. The "mpls/ldp/igp-sync" container in the global configuration is intentionally empty and is not required for feature activation. The goal of this empty container is to facilitate augmentation with additional parameters, e.g., timers.

## 2.7. ISO Parameters

As the IS-IS protocol is based on the ISO protocol suite, some ISO parameters may be required.

This module augments the "ietf-interfaces" module configuration to support selected ISO configuration parameters.

The "clns-mtu" parameter can be configured for an interface.

## 2.8. IP FRR

The YANG module defined in this document supports LFAs (Loop-Free Alternates) [RFC5286] and remote LFAs [RFC7490] as IP Fast Reroute (FRR) techniques. The "fast-reroute" container may be augmented by other models to support other IP FRR flavors (Maximally Redundant Trees (MRTs) as defined in [RFC7812], Topology Independent LFA (TI-LFA) FRR as defined in [SR-TI-LFA], etc.).

The current version of the model supports activation of LFAs and remote LFAs at the interface level only. The global "lfa" container is present but kept empty to allow augmentation with vendor-specific properties, e.g., policies.

Remote LFAs are considered an extension of LFAs. Remote LFAs cannot be enabled if LFAs are not enabled.

The "candidate-enabled" data leaf designates that an interface can be used as a backup.

## 2.9. Operational States

Operational states are defined in the "ietf-isis" module in various containers at various levels:

system-counters:

Provides statistical information about the global system.

interface:

Provides configuration state information for each interface.

adjacencies:

Provides state information about current IS-IS adjacencies.

spf-log:

Provides information about Shortest Path First (SPF) events for an IS-IS instance. This SHOULD be implemented as a wrapping buffer.

lsp-log:

Provides information about LSP events for an IS-IS instance (reception of an LSP or modification of a local LSP). This SHOULD be implemented as a wrapping buffer. The implementation MAY

optionally log LSP refreshes.

local-rib:  
Provides the IS-IS internal routing table.

database:  
Provides the contents of the current Link State Database (LSDB).

hostnames:  
Provides the system-ID-to-hostname mappings [RFC5301].

fast-reroute:  
Provides IP FRR state information.

### 3. RPC Operations

The "ietf-isis" module defines two RPC operations:

clear-database:  
Resets the contents of a particular IS-IS database and restarts database synchronization with all neighbors.

clear-adjacency:  
Restarts a particular set of IS-IS adjacencies.

### 4. Notifications

The "ietf-isis" module defines the following notifications:

database-overload:  
This notification is sent when the IS-IS node's overload condition changes.

lsp-too-large:  
This notification is sent when the system tries to propagate a PDU that is too large.

if-state-change:  
This notification is sent when an interface's state changes.

corrupted-lsp-detected:  
This notification is sent when the IS-IS node discovers that an LSP that was previously stored in the LSDB, i.e., local memory, has become corrupted.

attempt-to-exceed-max-sequence:  
This notification is sent when the system wraps the 32-bit sequence counter of an LSP.

id-len-mismatch:  
This notification is sent when a PDU with a different value for the system ID length is received.

max-area-addresses-mismatch:  
This notification is sent when a PDU with a different value for the Maximum Area Addresses is received.

own-lsp-purge:  
This notification is sent when the system receives a PDU with its own system ID and zero age.

sequence-number-skipped:  
This notification is sent when the system receives a PDU with its own system ID and different contents. The system has to reissue the LSP with a higher sequence number.

authentication-type-failure:

This notification is sent when the system receives a PDU with the wrong authentication type field.

authentication-failure:

This notification is sent when the system receives a PDU with the wrong authentication information.

version-skew:

This notification is sent when the system receives a PDU with a different protocol version number.

area-mismatch:

This notification is sent when the system receives a Hello PDU from an IS that does not share any area address.

rejected-adjacency:

This notification is sent when the system receives a Hello PDU from an IS but does not establish an adjacency for some reason.

protocols-supported-mismatch:

This notification is sent when the system receives a non-pseudonode LSP that has no matching protocol supported.

lsp-error-detected:

This notification is sent when the system receives an LSP with a parse error.

adjacency-state-change:

This notification is sent when an IS-IS adjacency moves to the "up" state or the "down" state.

lsp-received:

This notification is sent when an LSP is received.

lsp-generation:

This notification is sent when an LSP is regenerated.

## 5. Interactions with Other YANG Modules

The "isis" container augments the "/rt:routing/rt:control-plane-protocols/control-plane-protocol" container of the "ietf-routing" module [RFC8349] with IS-IS-specific parameters.

The "ietf-isis" module augments "/if:interfaces/if:interface" as defined by [RFC8343] with ISO-specific parameters.

Some IS-IS-specific route attributes are added to route objects in the "ietf-routing" module by augmenting "/rt:routing-state/rt:ribs/rt:rib/rt:routes/rt:route".

The module defined in this document uses some groupings from "ietf-keychain" [RFC8177].

The module reuses types from [RFC6991] and [RFC8294].

To support Bidirectional Forwarding Detection (BFD) for fast detection, the module relies on [RFC9314].

## 6. IS-IS YANG Module

The following specifications are referenced in this module:

[ISO-10589], [RFC1195], [RFC4090], [RFC5029], [RFC5120], [RFC5130], [RFC5286], [RFC5301], [RFC5302], [RFC5305], [RFC5307], [RFC5308], [RFC5443], [RFC5880], [RFC5881], [RFC6119], [RFC6232], [RFC6241], [RFC6991], [RFC7490], [RFC7794], [RFC7917], [RFC7981], [RFC8177],

[RFC8294], [RFC8342], [RFC8343], [RFC8349], [RFC8405], [RFC8570], [RFC8706], and [RFC9314].

```
<CODE BEGINS> file "ietf-isis@2022-10-19.yang"
module ietf-isis {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-isis";
  prefix isis;

  import ietf-routing {
    prefix rt;
    reference
      "RFC 8349: A YANG Data Model for Routing Management
      (NMDA Version)";
  }
  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 6991: Common YANG Data Types";
  }
  import ietf-yang-types {
    prefix yang;
    reference
      "RFC 6991: Common YANG Data Types";
  }
  import ietf-interfaces {
    prefix if;
    reference
      "RFC 8343: A YANG Data Model for Interface Management";
  }
  import ietf-key-chain {
    prefix key-chain;
    reference
      "RFC 8177: YANG Data Model for Key Chains";
  }
  import ietf-routing-types {
    prefix rt-types;
    reference
      "RFC 8294: Common YANG Data Types for the Routing Area";
  }
  import iana-routing-types {
    prefix iana-rt-types;
    reference
      "RFC 8294: Common YANG Data Types for the Routing Area";
  }
  import ietf-bfd-types {
    prefix bfd-types;
    reference
      "RFC 9314: YANG Data Model for Bidirectional Forwarding
      Detection (BFD)";
  }

  organization
    "IETF LSR Working Group";
  contact
    "WG Web:  <https://datatracker.ietf.org/wg/lsr/>
    WG List:  <mailto:lsr@ietf.org>

    Editor:    Stephane Litkowski
               <mailto:slitkows.ietf@gmail.com>

    Author:    Derek Yeung
               <mailto:derek@arrcus.com>

    Author:    Acee Lindem
               <mailto:acee@cisco.com>
```

Author: Jeffrey Zhang  
<mailto:zzhang@juniper.net>

Author: Ladislav Lhotka  
<mailto:ladislav.lhotka@nic.cz>;

description

"This YANG module defines the generic configuration and operational states for the IS-IS protocol common to all vendor implementations. It is intended that the module will be extended by vendors to define vendor-specific IS-IS configuration parameters and policies - for example, route maps or route policies.

This YANG data model conforms to the Network Management Datastore Architecture (NMDA) as described in RFC 8342.

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 (RFC 2119) (RFC 8174) when, and only when, they appear in all capitals, as shown here.

Copyright (c) 2022 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 Revised 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 9130; see the RFC itself for full legal notices."

reference

"RFC 8342: Network Management Datastore Architecture (NMDA)";

```
revision 2022-10-19 {  
  description  
    "Initial revision.";  
  reference  
    "RFC 9130: YANG Data Model for the IS-IS Protocol";  
}
```

/\* Identities \*/

```
identity isis {  
  base rt:routing-protocol;  
  description  
    "Identity for the IS-IS routing protocol.";  
}
```

```
identity lsp-log-reason {  
  description  
    "Base identity for a Link State PDU (LSP)  
    change log reason.";  
}
```

```
identity refresh {  
  base lsp-log-reason;  
  description  
    "Identity used when the LSP log reason is that an LSP  
    refresh was received.";  
}
```

```

identity content-change {
    base lsp-log-reason;
    description
        "Identity used when the LSP log reason is
        a change in the contents of the LSP.";
}

identity frr-protection-method {
    description
        "Base identity for a Fast Reroute protection method.";
}

identity frr-protection-method-lfa {
    base frr-protection-method;
    description
        "Loop-Free Alternate as defined in RFC 5286.";
    reference
        "RFC 5286: Basic Specification for IP Fast Reroute:
        Loop-Free Alternates";
}

identity frr-protection-method-rlfa {
    base frr-protection-method;
    description
        "Remote Loop-Free Alternate as defined in RFC 7490.";
    reference
        "RFC 7490: Remote Loop-Free Alternate (LFA)
        Fast Reroute (FRR)";
}

identity frr-protection-method-rsvp-te {
    base frr-protection-method;
    description
        "RSVP-TE as defined in RFC 4090.";
    reference
        "RFC 4090: Fast Reroute Extensions to RSVP-TE for
        LSP Tunnels";
}

identity frr-protection-available-type {
    description
        "Base identity for Fast Reroute protection types
        provided by an alternate path.";
}

identity frr-protection-available-node-type {
    base frr-protection-available-type;
    description
        "Node protection is provided by the alternate.";
}

identity frr-protection-available-link-type {
    base frr-protection-available-type;
    description
        "Link protection is provided by the alternate.";
}

identity frr-protection-available-srlg-type {
    base frr-protection-available-type;
    description
        "Shared Risk Link Group (SRLG) protection is provided by
        the alternate.";
}

identity frr-protection-available-downstream-type {
    base frr-protection-available-type;
}

```

```

    description
        "The alternate is downstream of the node in the path.";
}

identity frr-protection-available-other-type {
    base frr-protection-available-type;
    description
        "The level of protection is unknown.";
}

identity frr-alternate-type {
    description
        "Base identity for the IP Fast Reroute alternate type.";
}

identity frr-alternate-type-equal-cost {
    base frr-alternate-type;
    description
        "ECMP-based alternate.";
}

identity frr-alternate-type-lfa {
    base frr-alternate-type;
    description
        "LFA-based alternate.";
}

identity frr-alternate-type-remote-lfa {
    base frr-alternate-type;
    description
        "Remote-LFA-based alternate.";
}

identity frr-alternate-type-tunnel {
    base frr-alternate-type;
    description
        "Tunnel-based alternate (such as RSVP-TE or GRE).";
}

identity frr-alternate-mrt {
    base frr-alternate-type;
    description
        "MRT-based alternate.";
}

identity frr-alternate-tilfa {
    base frr-alternate-type;
    description
        "TI-LFA-based alternate.";
}

identity frr-alternate-other {
    base frr-alternate-type;
    description
        "Other type of alternate.";
}

identity unidirectional-link-delay-subtlv-flag {
    description
        "Base identity for the flag corresponding to the
        Unidirectional Link Delay sub-TLV as defined in RFC 8570.";
    reference
        "RFC 8570: IS-IS Traffic Engineering (TE) Metric Extensions";
}

identity unidirectional-link-delay-subtlv-a-flag {

```

```

base unidirectional-link-delay-subtlv-flag;
description
    "The 'A' bit field represents the Anomalous (A) bit.
    The A bit is set when the measured value of
    this parameter exceeds its configured
    maximum threshold.
    The A bit is cleared when the measured value
    falls below its configured reuse threshold.
    If the A bit is clear,
    the value represents steady-state link performance.";
}

identity min-max-unidirectional-link-delay-subtlv-flag {
    description
        "Base identity for the flag corresponding to the Min/Max
        Unidirectional Link Delay sub-TLV as defined in RFC 8570.";
    reference
        "RFC 8570: IS-IS Traffic Engineering (TE) Metric Extensions";
}

identity min-max-unidirectional-link-delay-subtlv-a-flag {
    base min-max-unidirectional-link-delay-subtlv-flag;
    description
        "The 'A' bit field represents the Anomalous (A) bit.
        The A bit is set when the measured value of
        this parameter exceeds its configured
        maximum threshold.
        The A bit is cleared when the measured value
        falls below its configured reuse threshold.
        If the A bit is clear,
        the value represents steady-state link performance.";
}

identity unidirectional-link-loss-subtlv-flag {
    description
        "Base identity for the flag corresponding to the
        Unidirectional Link Loss sub-TLV as defined in RFC 8570.";
    reference
        "RFC 8570: IS-IS Traffic Engineering (TE) Metric Extensions";
}

identity unidirectional-link-loss-subtlv-a-flag {
    base unidirectional-link-loss-subtlv-flag;
    description
        "The 'A' bit field represents the Anomalous (A) bit.
        The A bit is set when the measured value of
        this parameter exceeds its configured
        maximum threshold.
        The A bit is cleared when the measured value
        falls below its configured reuse threshold.
        If the A bit is clear,
        the value represents steady-state link performance.";
}

identity tlv229-flag {
    description
        "Base identity for the flag corresponding to TLV 229
        (M-Topologies) as defined in RFC 5120.";
    reference
        "RFC 5120: M-ISIS: Multi Topology (MT) Routing in
        Intermediate System to Intermediate Systems (IS-ISs)";
}

identity tlv229-overload-flag {
    base tlv229-flag;
    description

```

```

        "If set, the originator is overloaded
        and must be avoided in the path calculation.";
    }

identity tlv229-attached-flag {
    base tlv229-flag;
    description
        "If set, the originator is attached to
        another area using the referred metric.";
}

identity router-capability-flag {
    description
        "Base identity for the flag corresponding to the
        Router Capability TLV as defined in RFC 7981.";
    reference
        "RFC 7981: IS-IS Extensions for Advertising Router
        Information";
}

identity router-capability-flooding-flag {
    base router-capability-flag;
    description
        "Quote from RFC 7981:
        'If the S bit is set(1), the IS-IS Router CAPABILITY TLV
        MUST be flooded across the entire routing domain. If the
        S bit is not set(0), the TLV MUST NOT be leaked between
        levels. This bit MUST NOT be altered during the TLV
        leaking.'";
}

identity router-capability-down-flag {
    base router-capability-flag;
    description
        "Quote from RFC 7981:
        'When the IS-IS Router CAPABILITY TLV is leaked from
        Level 2 (L2) to Level 1 (L1), the D bit MUST be set.
        Otherwise, this bit MUST be clear. IS-IS Router
        CAPABILITY TLVs with the D bit set MUST NOT be leaked from
        Level 1 to Level 2. This is to prevent TLV looping.'";
}

identity lsp-flag {
    description
        "Base identity for LSP attributes as defined in ISO 10589.";
    reference
        "ISO 10589: Intermediate System to Intermediate System
        intra-domain routeing information exchange protocol
        for use in conjunction with the protocol for providing
        the connectionless-mode network service (ISO 8473)";
}

identity lsp-partitioned-flag {
    base lsp-flag;
    description
        "Originator partition repair supported.";
}

identity lsp-attached-error-metric-flag {
    base lsp-flag;
    description
        "Set when the originator is attached to
        another area using the error metric.";
}

identity lsp-attached-delay-metric-flag {

```

```

    base lsp-flag;
    description
        "Set when the originator is attached to
        another area using the delay metric.";
}

identity lsp-attached-expense-metric-flag {
    base lsp-flag;
    description
        "Set when the originator is attached to
        another area using the expense metric.";
}

identity lsp-attached-default-metric-flag {
    base lsp-flag;
    description
        "Set when the originator is attached to
        another area using the default metric.";
}

identity lsp-overload-flag {
    base lsp-flag;
    description
        "If set, the originator is overloaded
        and must be avoided in the path calculation.";
}

identity lsp-l1-system-flag {
    base lsp-flag;
    description
        "Set when the Intermediate System has an L1 type.";
}

identity lsp-l2-system-flag {
    base lsp-flag;
    description
        "Set when the Intermediate System has an L2 type.";
}

/* Feature definitions */

feature osi-interface {
    description
        "Support of OSI-specific parameters on an interface.";
}

feature poi-tlv {
    description
        "Support of the Purge Originator Identification (POI) TLV.";
    reference
        "RFC 6232: Purge Originator Identification TLV for IS-IS";
}

feature ietf-spf-delay {
    description
        "Support for the IETF SPF delay algorithm.";
    reference
        "RFC 8405: Shortest Path First (SPF) Back-Off Delay Algorithm
        for Link-State IGP";
}

feature bfd {
    description
        "Support for detection of IS-IS neighbor reachability
        via BFD.";
    reference

```

```

        "RFC 5880: Bidirectional Forwarding Detection (BFD)
        RFC 5881: Bidirectional Forwarding Detection (BFD)
        for IPv4 and IPv6 (Single Hop)";
    }

feature key-chain {
    description
        "Support of key chains for authentication.";
    reference
        "RFC 8177: YANG Data Model for Key Chains";
}

feature node-flag {
    description
        "Support for node flags for IS-IS prefixes.";
    reference
        "RFC 7794: IS-IS Prefix Attributes for Extended IPv4 and IPv6
        Reachability";
}

feature node-tag {
    description
        "Support for node administrative tags for IS-IS
        routing instances.";
    reference
        "RFC 7917: Advertising Node Administrative Tags in IS-IS";
}

feature ldp-igp-sync {
    description
        "Support for LDP IGP synchronization.";
    reference
        "RFC 5443: LDP IGP Synchronization";
}

feature fast-reroute {
    description
        "Support for IP Fast Reroute (IP FRR).";
}

feature nsr {
    description
        "Support for Non-Stop-Routing (NSR). The IS-IS NSR feature
        allows a router with redundant control-plane capability
        (e.g., dual Route Processor (RP) cards) to maintain its
        state and adjacencies during planned and unplanned
        IS-IS instance restarts. It differs from graceful restart
        or Non-Stop Forwarding (NSF) in that no protocol signaling
        or assistance from adjacent IS-IS neighbors is required to
        recover control-plane state.";
}

feature lfa {
    description
        "Support for Loop-Free Alternates (LFAs).";
    reference
        "RFC 5286: Basic Specification for IP Fast Reroute:
        Loop-Free Alternates";
}

feature remote-lfa {
    description
        "Support for remote LFAs (R-LFAs).";
    reference
        "RFC 7490: Remote Loop-Free Alternate (LFA)
        Fast Reroute (FRR)";
}

```

```

}

feature overload-max-metric {
    description
        "Support of overload by setting all links to the maximum
        link metric. In IS-IS, the overload bit is usually used to
        signal that a node cannot be used as a transit node. The
        'overload-max-metric' feature provides similar behavior,
        also setting all the link metrics to MAX_METRIC.";
}

feature prefix-tag {
    description
        "Support for 32-bit prefix tags.";
    reference
        "RFC 5130: A Policy Control Mechanism in IS-IS Using
        Administrative Tags";
}

feature prefix-tag64 {
    description
        "Support for 64-bit prefix tags.";
    reference
        "RFC 5130: A Policy Control Mechanism in IS-IS Using
        Administrative Tags";
}

feature auto-cost {
    description
        "Support for an IS-IS interface metric computation
        according to a reference bandwidth.";
}

feature te-rid {
    description
        "Traffic Engineering router ID.";
    reference
        "RFC 5305: IS-IS Extensions for Traffic Engineering
        RFC 6119: IPv6 Traffic Engineering in IS-IS";
}

feature max-ecmp {
    description
        "Sets the maximum number of ECMP paths.";
}

feature multi-topology {
    description
        "Support for Multi-Topology (MT) Routing.";
    reference
        "RFC 5120: M-ISIS: Multi Topology (MT) Routing in
        Intermediate System to Intermediate Systems (IS-ISs)";
}

feature nlpid-control {
    description
        "Support for the advertisement of a Network Layer
        Protocol Identifier within an IS-IS configuration.";
}

feature graceful-restart {
    description
        "Support for IS-IS graceful restart.";
    reference
        "RFC 8706: Restart Signaling for IS-IS";
}

```

```

feature lsp-refresh {
    description
        "Configuration of the LSP refresh interval.";
}

feature maximum-area-addresses {
    description
        "Support for 'maximum-area-addresses' configuration.";
}

feature admin-control {
    description
        "Administrative control of the protocol state.";
}

/* Type definitions */

typedef circuit-id {
    type uint8;
    description
        "This type defines the circuit ID
        associated with an interface.";
}

typedef extended-circuit-id {
    type uint32;
    description
        "This type defines the extended circuit ID
        associated with an interface.";
}

typedef interface-type {
    type enumeration {
        enum broadcast {
            description
                "Broadcast interface type.";
        }
        enum point-to-point {
            description
                "Point-to-point interface type.";
        }
    }
    description
        "This type defines the type of adjacency
        to be established for the interface.
        'interface-type' determines the type
        of Hello message that is used.";
}

typedef level {
    type enumeration {
        enum level-1 {
            description
                "This enum indicates L1-only capability.";
        }
        enum level-2 {
            description
                "This enum indicates L2-only capability.";
        }
        enum level-all {
            description
                "This enum indicates capability for both levels.";
        }
    }
}
default "level-all";

```

```

    description
        "This type defines the IS-IS level of an object.";
}

typedef adj-state-type {
    type enumeration {
        enum up {
            description
                "This state indicates that the adjacency is established.";
        }
        enum down {
            description
                "This state indicates that the adjacency is
                NOT established.";
        }
        enum init {
            description
                "This state indicates that the adjacency is being
                established.";
        }
        enum failed {
            description
                "This state indicates that the adjacency has failed.";
        }
    }
    description
        "This type defines the states of an adjacency.";
}

typedef if-state-type {
    type enumeration {
        enum up {
            description
                "'up' state.";
        }
        enum down {
            description
                "'down' state.";
        }
    }
    description
        "This type defines the state of an interface.";
}

typedef level-number {
    type uint8 {
        range "1 .. 2";
    }
    description
        "This type defines the current IS-IS level.";
}

typedef lsp-id {
    type string {
        pattern '[0-9A-Fa-f]{4}\.[0-9A-Fa-f]{4}\.[0-9A-Fa-f]'
            + '{4}\.[0-9][0-9]-[0-9][0-9]';
    }
    description
        "This type defines the IS-IS LSP ID format using a
        pattern. An example LSP ID is '0143.0438.AEF0.02-01'.";
}

typedef area-address {
    type string {
        pattern '[0-9A-Fa-f]{2}(\.[0-9A-Fa-f]{4}){0,6}';
    }
}

```

```

    description
        "This type defines the area address format.";
}

typedef snpa {
    type string {
        length "0 .. 20";
    }
    description
        "This type defines the Subnetwork Point of Attachment (SNPA)
        format. The SNPA should be encoded according to the rules
        specified for the particular type of subnetwork being used.
        As an example, for an Ethernet subnetwork, the SNPA is
        encoded as a Media Access Control (MAC) address, such as
        '00aa.bbccc.ddee'.";
}

typedef system-id {
    type string {
        pattern '[0-9A-Fa-f]{4}\.[0-9A-Fa-f]{4}\.[0-9A-Fa-f]{4}';
    }
    description
        "This type defines the IS-IS system ID by using a pattern.
        An example system ID is '0143.0438.AEF0'.";
}

typedef extended-system-id {
    type string {
        pattern '[0-9A-Fa-f]{4}\.[0-9A-Fa-f]{4}\.[0-9A-Fa-f]{4}\.'
            + '[0-9][0-9]';
    }
    description
        "This type defines the IS-IS system ID using a pattern.
        'extended-system-id' contains the pseudonode number
        in addition to the system ID.
        An example extended system ID is '0143.0438.AEF0.00'.";
}

typedef wide-metric {
    type uint32 {
        range "0 .. 16777215";
    }
    description
        "This type defines the wide-style format of an IS-IS metric.";
}

typedef std-metric {
    type uint8 {
        range "0 .. 63";
    }
    description
        "This type defines the old-style format of the IS-IS metric.";
}

typedef mesh-group-state {
    type enumeration {
        enum mesh-inactive {
            description
                "The interface is not part of a mesh group.";
        }
        enum mesh-set {
            description
                "The interface is part of a mesh group.";
        }
        enum mesh-blocked {
            description

```

```

        "LSPs must not be flooded over this interface.";
    }
}
description
    "This type describes the mesh group state of an interface.";
}

/* Grouping for notifications */

grouping notification-instance-hdr {
    description
        "Instance-specific IS-IS notification data grouping.";
    leaf routing-protocol-name {
        type leafref {
            path "/rt:routing/rt:control-plane-protocols/"
                + "rt:control-plane-protocol/rt:name";
        }
        description
            "Name of the IS-IS instance.";
    }
    leaf isis-level {
        type level;
        description
            "IS-IS level of the instance.";
    }
}

grouping notification-interface-hdr {
    description
        "Interface-specific IS-IS notification data grouping.";
    leaf interface-name {
        type if:interface-ref;
        description
            "IS-IS interface name.";
    }
    leaf interface-level {
        type level;
        description
            "IS-IS level of the interface.";
    }
    leaf extended-circuit-id {
        type extended-circuit-id;
        description
            "Extended circuit ID of the interface.";
    }
}

/* Groupings for IP Fast Reroute */

grouping instance-fast-reroute-config {
    description
        "This group defines the global configuration of IP
        Fast Reroute (FRR).";
    container fast-reroute {
        if-feature "fast-reroute";
        description
            "This container may be augmented with global
            parameters for IP FRR.";
        container lfa {
            if-feature "lfa";
            description
                "This container may be augmented with
                global parameters for Loop-Free Alternates (LFAs).
                The creation of this container has no effect on
                LFA activation.";
        }
    }
}

```

```

    }
}

grouping interface-lfa-config {
    leaf candidate-enabled {
        type boolean;
        default "true";
        description
            "Enables the interface to be used as a backup.";
    }
    leaf enabled {
        type boolean;
        default "false";
        description
            "Activates the LFA. Per-prefix LFA computation is assumed.";
    }
    container remote-lfa {
        if-feature "remote-lfa";
        leaf enabled {
            type boolean;
            default "false";
            description
                "Activates the remote LFA (R-LFA).";
        }
        description
            "Remote LFA configuration.";
    }
    description
        "Grouping for LFA interface configuration.";
}

```

```

grouping interface-fast-reroute-config {
    description
        "This group defines the interface configuration of IP FRR.";
    container fast-reroute {
        if-feature "fast-reroute";
        container lfa {
            if-feature "lfa";
            uses interface-lfa-config;
            container level-1 {
                uses interface-lfa-config;
                description
                    "LFA level-1 configuration.";
            }
            container level-2 {
                uses interface-lfa-config;
                description
                    "LFA level-2 configuration.";
            }
            description
                "LFA configuration.";
        }
        description
            "Interface IP FRR configuration.";
    }
}

```

```

grouping instance-fast-reroute-state {
    description
        "IP FRR state data grouping.";
    container protected-routes {
        config false;
        list address-family-stats {
            key "address-family prefix alternate";
            leaf address-family {
                type iana-rt-types:address-family;
            }
        }
    }
}

```

```

    description
        "Address family.";
}
leaf prefix {
    type inet:ip-prefix;
    description
        "Protected prefix.";
}
leaf alternate {
    type inet:ip-address;
    description
        "Alternate next hop for the prefix.";
}
leaf alternate-type {
    type identityref {
        base frr-alternate-type;
    }
    description
        "Type of alternate.";
}
leaf best {
    type boolean;
    description
        "Set when the alternate is the preferred alternate;
        clear otherwise.";
}
leaf non-best-reason {
    type string {
        length "1..255";
    }
    description
        "Information field that explains why the alternate
        is not the best alternate. The length should be
        limited to 255 Unicode characters. The expected format
        is a single line of text.";
}
container protection-available {
    leaf-list protection-types {
        type identityref {
            base frr-protection-available-type;
        }
    }
    description
        "This list contains a set of protection
        types defined as identities.
        An identity must be added for each type of
        protection provided by the alternate.
        As an example, if an alternate provides
        SRLG, node, and link protection, three
        identities must be added in this list:
        one for SRLG protection, one for node
        protection, and one for link protection.";
}
description
    "Protection types provided by the alternate.";
}
leaf alternate-metric-1 {
    type uint32;
    description
        "Metric from the Point of Local Repair (PLR) to the
        destination through the alternate path.";
}
leaf alternate-metric-2 {
    type uint32;
    description
        "Metric from the PLR to the alternate node.";
}
}

```

```

    leaf alternate-metric-3 {
        type uint32;
        description
            "Metric from the alternate node to the destination.";
    }
    description
        "Per-address-family protected prefix statistics.";
}
description
    "List of prefixes that are protected.";
}
container unprotected-routes {
    config false;
    list prefixes {
        key "address-family prefix";
        leaf address-family {
            type iana-rt-types:address-family;
            description
                "Address family.";
        }
        leaf prefix {
            type inet:ip-prefix;
            description
                "Unprotected prefix.";
        }
    }
    description
        "Per-address-family unprotected prefix statistics.";
}
description
    "List of prefixes that are not protected.";
}
list protection-statistics {
    key "frr-protection-method";
    config false;
    leaf frr-protection-method {
        type identityref {
            base frr-protection-method;
        }
        description
            "Protection method used.";
    }
}
list address-family-stats {
    key "address-family";
    leaf address-family {
        type iana-rt-types:address-family;
        description
            "Address family.";
    }
    leaf total-routes {
        type yang:gauge32;
        description
            "Total prefixes.";
    }
    leaf unprotected-routes {
        type yang:gauge32;
        description
            "Total prefixes that are not protected.";
    }
    leaf protected-routes {
        type yang:gauge32;
        description
            "Total prefixes that are protected.";
    }
}
leaf link-protected-routes {
    type yang:gauge32;
    description

```

```

        "Total prefixes that are link protected.";
    }
    leaf node-protected-routes {
        type yang:gauge32;
        description
            "Total prefixes that are node protected.";
    }
    description
        "Per-address-family protected prefix statistics.";
}
description
    "Global protection statistics.";
}
}

/* Routing table and local Routing Information Base (RIB)
   groupings */

grouping local-rib {
    description
        "Local RIB: RIB for routes computed by the local IS-IS
        routing instance.";
    container local-rib {
        config false;
        description
            "Local RIB.";
        list route {
            key "prefix";
            description
                "Routes.";
            leaf prefix {
                type inet:ip-prefix;
                description
                    "Destination prefix.";
            }
            container next-hops {
                description
                    "Next hops for the route.";
                list next-hop {
                    key "next-hop";
                    description
                        "List of next hops for the route.";
                    leaf outgoing-interface {
                        type if:interface-ref;
                        description
                            "Name of the outgoing interface.";
                    }
                    leaf next-hop {
                        type inet:ip-address;
                        description
                            "Next-hop address.";
                    }
                }
            }
        }
        leaf metric {
            type uint32;
            description
                "Metric for this route.";
        }
        leaf level {
            type level-number;
            description
                "Level number for this route.";
        }
        leaf route-tag {
            type uint32;

```

```

        description
            "Route tag for this route.";
    }
}
}

grouping route-content {
    description
        "IS-IS protocol-specific route properties grouping.";
    leaf metric {
        type uint32;
        description
            "IS-IS metric of a route.";
    }
    leaf-list tag {
        type uint64;
        description
            "List of tags associated with the route. This list
            provides a consolidated view of both 32-bit and 64-bit
            tags (RFC 5130) available for the prefix.";
        reference
            "RFC 5130: A Policy Control Mechanism in IS-IS Using
            Administrative Tags";
    }
    leaf route-type {
        type enumeration {
            enum l2-intra-area {
                description
                    "Level-2 internal route. As per RFC 5302,
                    the prefix is directly connected to the
                    advertising router. It cannot be
                    distinguished from an L1->L2 inter-area
                    route.";
                reference
                    "RFC 5302: Domain-Wide Prefix Distribution with
                    Two-Level IS-IS";
            }
            enum l1-intra-area {
                description
                    "Level-1 internal route. As per RFC 5302,
                    the prefix is directly connected to the
                    advertising router.";
            }
            enum l2-external {
                description
                    "Level-2 external route. As per RFC 5302,
                    such a route is learned from other IGPs.
                    It cannot be distinguished from an L1->L2
                    inter-area external route.";
            }
            enum l1-external {
                description
                    "Level-1 external route. As per RFC 5302,
                    such a route is learned from other IGPs.";
            }
            enum l1-inter-area {
                description
                    "These prefixes are learned via L2 routing.";
            }
            enum l1-inter-area-external {
                description
                    "These prefixes are learned via L2 routing
                    towards a level-2 external route.";
            }
        }
    }
}

```

```

        description
            "IS-IS route type.";
    }
}

/* Grouping definitions for configuration and operational states */

grouping adjacency-state {
    container adjacencies {
        config false;
        list adjacency {
            leaf neighbor-sys-type {
                type level;
                description
                    "Level capability of the neighboring system.";
            }
            leaf neighbor-sysid {
                type system-id;
                description
                    "The system ID of the neighbor.";
            }
            leaf neighbor-extended-circuit-id {
                type extended-circuit-id;
                description
                    "The circuit ID of the neighbor.";
            }
            leaf neighbor-snpa {
                type snpa;
                description
                    "The SNPA of the neighbor.";
            }
            leaf usage {
                type level;
                description
                    "Defines the level(s) activated for the adjacency.
                     On a point-to-point link, this might be level 1 and
                     level 2, but on a LAN, the usage will be level 1
                     between neighbors at level 1 or level 2 between
                     neighbors at level 2.";
            }
            leaf hold-timer {
                type rt-types:timer-value-seconds16;
                units "seconds";
                description
                    "The holding time (in seconds) for this adjacency.
                     This value is based on received Hello PDUs and the
                     elapsed time since receipt.";
            }
            leaf neighbor-priority {
                type uint8 {
                    range "0 .. 127";
                }
                description
                    "Priority of the neighboring IS for becoming the
                     Designated Intermediate System (DIS).";
            }
            leaf lastuptime {
                type yang:timestamp;
                description
                    "When the adjacency most recently entered the
                     'up' state, measured in hundredths of a
                     second since the last reinitialization of
                     the network management subsystem.
                     The value is 0 if the adjacency has never
                     been in the 'up' state.";
            }
        }
    }
}

```

```

        leaf state {
            type adj-state-type;
            description
                "This leaf describes the state of the interface.";
        }
        description
            "List of operational adjacencies.";
    }
    description
        "This container lists the adjacencies of
        the local node.";
}
description
    "Adjacency state.";
}

grouping admin-control {
    leaf enabled {
        if-feature "admin-control";
        type boolean;
        default "true";
        description
            "Enables or disables the protocol.";
    }
    description
        "Grouping for administrative control.";
}

grouping ietf-spf-delay {
    leaf initial-delay {
        type rt-types:timer-value-milliseconds;
        units "msec";
        default "50";
        description
            "Delay used while in the QUIET state (milliseconds).";
    }
    leaf short-delay {
        type rt-types:timer-value-milliseconds;
        units "msec";
        default "200";
        description
            "Delay used while in the SHORT_WAIT state (milliseconds).";
    }
    leaf long-delay {
        type rt-types:timer-value-milliseconds;
        units "msec";
        default "5000";
        description
            "Delay used while in the LONG_WAIT state (milliseconds).";
    }
    leaf hold-down {
        type rt-types:timer-value-milliseconds;
        units "msec";
        default "10000";
        description
            "This timer value defines the period without any changes
            for the IGP to be considered stable (in milliseconds).";
    }
    leaf time-to-learn {
        type rt-types:timer-value-milliseconds;
        units "msec";
        default "500";
        description
            "Duration used to learn all the IGP events
            related to a single network event (milliseconds).";
    }
}

```

```

leaf current-state {
  type enumeration {
    enum quiet {
      description
        "QUIET state.";
    }
    enum short-wait {
      description
        "SHORT_WAIT state.";
    }
    enum long-wait {
      description
        "LONG_WAIT state.";
    }
  }
  config false;
  description
    "Current SPF Back-Off algorithm state.";
}
leaf remaining-time-to-learn {
  type rt-types:timer-value-milliseconds;
  units "msec";
  config false;
  description
    "Remaining time until the time-to-learn timer fires.";
}
leaf remaining-hold-down {
  type rt-types:timer-value-milliseconds;
  units "msec";
  config false;
  description
    "Remaining time until the hold-down timer fires.";
}
leaf last-event-received {
  type yang:timestamp;
  config false;
  description
    "Time of the last IGP event received.";
}
leaf next-spf-time {
  type yang:timestamp;
  config false;
  description
    "Time when the next SPF has been scheduled.";
}
leaf last-spf-time {
  type yang:timestamp;
  config false;
  description
    "Time of the last SPF computation.";
}
description
  "Grouping for IETF SPF delay configuration and state.";
reference
  "RFC 8405: Shortest Path First (SPF) Back-Off Delay Algorithm
  for Link-State IGPs";
}

grouping node-tag-config {
  description
    "IS-IS node tag configuration state.";
  container node-tags {
    if-feature "node-tag";
    list node-tag {
      key "tag";
      leaf tag {

```

```

        type uint32;
        description
            "Node tag value.";
    }
    description
        "List of tags.";
}
description
    "Container for node administrative tags.";
}
}

grouping authentication-global-cfg {
    choice authentication-type {
        case key-chain {
            if-feature "key-chain";
            leaf key-chain {
                type key-chain:key-chain-ref;
                description
                    "Reference to a key chain.";
            }
        }
        case password {
            leaf key {
                type string;
                description
                    "This leaf specifies the authentication key. The
                    length of the key may be dependent on the
                    cryptographic algorithm.";
            }
            leaf crypto-algorithm {
                type identityref {
                    base key-chain:crypto-algorithm;
                }
                description
                    "Cryptographic algorithm associated with a key.";
            }
        }
    }
    description
        "Choice of authentication.";
}
description
    "Grouping for global authentication configuration.";
}

grouping metric-type-global-cfg {
    leaf value {
        type enumeration {
            enum wide-only {
                description
                    "Advertises the new metric style only (RFC 5305).";
                reference
                    "RFC 5305: IS-IS Extensions for Traffic Engineering";
            }
            enum old-only {
                description
                    "Advertises the old metric style only (RFC 1195).";
                reference
                    "RFC 1195: Use of OSI IS-IS for routing in TCP/IP and
                    dual environments";
            }
            enum both {
                description
                    "Advertises both metric styles.";
            }
        }
    }
}

```

```

    description
        "Type of metric to be generated:

        - 'wide-only' means that only a new metric style
          is generated.
        - 'old-only' means that only an old metric style
          is generated.
        - 'both' means that both are advertised.

        This leaf only affects IPv4 metrics.";
    }
    description
        "Grouping for global metric style configuration.";
}

grouping metric-type-global-cfg-with-default {
    leaf value {
        type enumeration {
            enum wide-only {
                description
                    "Advertises the new metric style only (RFC 5305).";
                reference
                    "RFC 5305: IS-IS Extensions for Traffic Engineering";
            }
            enum old-only {
                description
                    "Advertises the old metric style only (RFC 1195).";
                reference
                    "RFC 1195: Use of OSI IS-IS for routing in TCP/IP and
                     dual environments";
            }
            enum both {
                description
                    "Advertises both metric styles.";
            }
        }
        default "wide-only";
        description
            "Type of metric to be generated:

            - 'wide-only' means that only a new metric style
              is generated.
            - 'old-only' means that only an old metric style
              is generated.
            - 'both' means that both are advertised.

            This leaf only affects IPv4 metrics.";
    }
    description
        "Grouping for global metric style configuration.";
}

grouping default-metric-global-cfg {
    leaf value {
        type wide-metric;
        description
            "Value of the metric.";
    }
    description
        "Global default metric configuration grouping.";
}

grouping default-metric-global-cfg-with-default {
    leaf value {
        type wide-metric;
        default "10";
    }

```

```

        description
            "Value of the metric.";
    }
    description
        "Global default metric configuration grouping.";
}

grouping overload-global-cfg {
    leaf status {
        type boolean;
        default "false";
        description
            "This leaf specifies the overload status.";
    }
    description
        "Grouping for overload bit configuration.";
}

grouping overload-max-metric-global-cfg {
    leaf timeout {
        type rt-types:timer-value-seconds16;
        units "seconds";
        description
            "Timeout (in seconds) of the overload condition.";
    }
    description
        "Overload maximum metric configuration grouping.";
}

grouping route-preference-global-cfg {
    choice granularity {
        case detail {
            leaf internal {
                type uint8;
                description
                    "Protocol preference for internal routes.";
            }
            leaf external {
                type uint8;
                description
                    "Protocol preference for external routes.";
            }
        }
        case coarse {
            leaf default {
                type uint8;
                description
                    "Protocol preference for all IS-IS routes.";
            }
        }
    }
    description
        "Choice for implementation of route preference.";
}
description
    "Global route preference grouping.";
}

grouping hello-authentication-cfg {
    choice authentication-type {
        case key-chain {
            if-feature "key-chain";
            leaf key-chain {
                type key-chain:key-chain-ref;
                description
                    "Reference to a key chain.";
            }
        }
    }
}

```

```

    }
    case password {
        leaf key {
            type string;
            description
                "Authentication key specification. The length of the
                key may be dependent on the cryptographic algorithm.";
        }
        leaf crypto-algorithm {
            type identityref {
                base key-chain:crypto-algorithm;
            }
            description
                "Cryptographic algorithm associated with a key.";
        }
    }
    description
        "Choice of authentication.";
}
description
    "Grouping for Hello authentication.";
}

grouping hello-interval-cfg {
    leaf value {
        type rt-types:timer-value-seconds16;
        units "seconds";
        description
            "Interval (in seconds) between successive Hello
            messages.";
    }
    description
        "Interval between Hello messages.";
}

grouping hello-interval-cfg-with-default {
    leaf value {
        type rt-types:timer-value-seconds16;
        units "seconds";
        default "10";
        description
            "Interval (in seconds) between successive Hello
            messages.";
    }
    description
        "Interval between Hello messages.";
}

grouping hello-multiplier-cfg {
    leaf value {
        type uint16;
        description
            "Number of missed Hello messages prior to
            declaring the adjacency down.";
    }
    description
        "Grouping for the number of missed Hello messages prior to
        declaring the adjacency down.";
}

grouping hello-multiplier-cfg-with-default {
    leaf value {
        type uint16;
        default "3";
        description
            "Number of missed Hello messages prior to

```

```

        declaring the adjacency down.";
    }
    description
        "Grouping for the number of missed Hello messages prior to
        declaring the adjacency down.";
}

grouping priority-cfg {
    leaf value {
        type uint8 {
            range "0 .. 127";
        }
        description
            "Priority of the interface for DIS election.";
    }
    description
        "Interface DIS election priority grouping.";
}

grouping priority-cfg-with-default {
    leaf value {
        type uint8 {
            range "0 .. 127";
        }
        default "64";
        description
            "Priority of the interface for DIS election.";
    }
    description
        "Interface DIS election priority grouping.";
}

grouping metric-cfg {
    leaf value {
        type wide-metric;
        description
            "Metric value.";
    }
    description
        "Interface metric grouping.";
}

grouping metric-cfg-with-default {
    leaf value {
        type wide-metric;
        default "10";
        description
            "Metric value.";
    }
    description
        "Interface metric grouping.";
}

grouping metric-parameters {
    container metric-type {
        uses metric-type-global-cfg-with-default;
        container level-1 {
            uses metric-type-global-cfg;
            description
                "Configuration specific to level 1.";
        }
        container level-2 {
            uses metric-type-global-cfg;
            description
                "Configuration specific to level 2.";
        }
    }
}

```

```

    description
        "Metric style global configuration.";
}
container default-metric {
    uses default-metric-global-cfg-with-default;
    container level-1 {
        uses default-metric-global-cfg;
        description
            "Configuration specific to level 1.";
    }
    container level-2 {
        uses default-metric-global-cfg;
        description
            "Configuration specific to level 2.";
    }
    description
        "Default metric global configuration.";
}
container auto-cost {
    if-feature "auto-cost";
    description
        "Interface auto-cost configuration state.";
    leaf enabled {
        type boolean;
        description
            "Enables or disables interface auto-cost.";
    }
    leaf reference-bandwidth {
        when "../enabled = 'true'" {
            description
                "Only when auto-cost is enabled.";
        }
        type uint32 {
            range "1..4294967";
        }
        units "Mbits";
        description
            "Configures the reference bandwidth used to automatically
            determine the interface cost (Mbits). The cost is the
            reference bandwidth divided by the interface speed,
            with 1 being the minimum cost.";
    }
}
description
    "Grouping for global metric parameters.";
}

grouping high-availability-parameters {
    container graceful-restart {
        if-feature "graceful-restart";
        leaf enabled {
            type boolean;
            default "false";
            description
                "Enables graceful restart.";
        }
        leaf restart-interval {
            type rt-types:timer-value-seconds16;
            units "seconds";
            description
                "Interval (in seconds) to attempt graceful restart prior
                to failure.";
        }
        leaf helper-enabled {
            type boolean;
            default "true";
        }
    }
}

```

```

        description
            "Enables a local IS-IS router as a graceful restart
            helper.";
    }
    description
        "Configuration of graceful restart.";
}
container nsr {
    if-feature "nsr";
    description
        "Non-Stop Routing (NSR) configuration.";
    leaf enabled {
        type boolean;
        default "false";
        description
            "Enables or disables NSR.";
    }
}
description
    "Grouping for high-availability parameters.";
}

grouping authentication-parameters {
    container authentication {
        uses authentication-global-cfg;
        container level-1 {
            uses authentication-global-cfg;
            description
                "Configuration specific to level 1.";
        }
        container level-2 {
            uses authentication-global-cfg;
            description
                "Configuration specific to level 2.";
        }
    }
    description
        "Authentication global configuration for
        both LSPs and Sequence Number PDUs (SNPs).";
}
description
    "Grouping for authentication parameters.";
}

grouping address-family-parameters {
    container address-families {
        if-feature "nlpid-control";
        list address-family-list {
            key "address-family";
            leaf address-family {
                type iana-rt-types:address-family;
                description
                    "Address family.";
            }
            leaf enabled {
                type boolean;
                description
                    "Activates the address family.";
            }
        }
        description
            "List of address families and whether or not they
            are activated.";
    }
    description
        "Address family configuration.";
}
description

```

```

    "Grouping for address family parameters.";
}

grouping mpls-parameters {
    container mpls {
        container te-rid {
            if-feature "te-rid";
            description
                "Stable IS-IS router IP address used for Traffic
                Engineering.";
            leaf ipv4-router-id {
                type inet:ipv4-address;
                description
                    "Router ID value that would be used in TLV 134.";
            }
            leaf ipv6-router-id {
                type inet:ipv6-address;
                description
                    "Router ID value that would be used in TLV 140.";
            }
        }
        container ldp {
            container igp-sync {
                if-feature "ldp-igp-sync";
                description
                    "This container may be augmented with global
                    parameters for LDP IGP synchronization.";
            }
            description
                "LDP configuration.";
        }
        description
            "MPLS configuration.";
    }
    description
        "Grouping for MPLS global parameters.";
}

grouping lsp-parameters {
    leaf lsp-mtu {
        type uint16;
        units "bytes";
        default "1492";
        description
            "Maximum size of an LSP PDU in bytes.";
    }
    leaf lsp-lifetime {
        type uint16 {
            range "1..65535";
        }
        units "seconds";
        description
            "Lifetime of the router's LSPs in seconds.";
    }
    leaf lsp-refresh {
        if-feature "lsp-refresh";
        type rt-types:timer-value-seconds16;
        units "seconds";
        description
            "Refresh interval of the router's LSPs in seconds.";
    }
    leaf poi-tlv {
        if-feature "poi-tlv";
        type boolean;
        default "false";
        description

```

```

        "Enables the advertisement of the IS-IS Purge Originator
        Identification TLV.";
    }
    description
        "Grouping for LSP global parameters.";
}

grouping spf-parameters {
    container spf-control {
        leaf paths {
            if-feature "max-ecmp";
            type uint16 {
                range "1..65535";
            }
            description
                "Maximum number of Equal-Cost Multi-Path (ECMP) paths.";
        }
        container ietf-spf-delay {
            if-feature "ietf-spf-delay";
            uses ietf-spf-delay;
            description
                "IETF SPF delay algorithm configuration.";
        }
        description
            "SPF calculation control.";
    }
    description
        "Grouping for SPF global parameters.";
}

grouping instance-config {
    description
        "IS-IS global configuration grouping.";
    uses admin-control;
    leaf level-type {
        type level;
        default "level-all";
        description
            "Level of an IS-IS node.  Can be 'level-1', 'level-2', or
            'level-all'.";
    }
    leaf system-id {
        type system-id;
        description
            "System ID of the node.";
    }
    leaf maximum-area-addresses {
        if-feature "maximum-area-addresses";
        type uint8;
        default "3";
        description
            "Maximum areas supported.";
    }
    leaf-list area-address {
        type area-address;
        description
            "List of areas supported by the protocol instance.";
    }
    uses lsp-parameters;
    uses high-availability-parameters;
    uses node-tag-config;
    uses metric-parameters;
    uses authentication-parameters;
    uses address-family-parameters;
    uses mpls-parameters;
    uses spf-parameters;
}

```

```

uses instance-fast-reroute-config;
container preference {
    uses route-preference-global-cfg;
    description
        "Router preference configuration for IS-IS
        protocol instance route installation.";
}
container overload {
    uses overload-global-cfg;
    description
        "Router protocol instance overload state configuration.";
}
container overload-max-metric {
    if-feature "overload-max-metric";
    uses overload-max-metric-global-cfg;
    description
        "Router protocol instance overload maximum
        metric advertisement configuration.";
}
}

grouping instance-state {
    description
        "IS-IS instance operational state.";
    uses spf-log;
    uses lsp-log;
    uses hostname-db;
    uses lsdb;
    uses local-rib;
    uses system-counters;
    uses instance-fast-reroute-state;
    leaf discontinuity-time {
        type yang:date-and-time;
        description
            "The time of the most recent occasion at which any one
            or more of this IS-IS instance's counters suffered a
            discontinuity.  If no such discontinuities have occurred
            since the IS-IS instance was last reinitialized, then
            this node contains the time the IS-IS instance was
            reinitialized, which normally occurs when it was
            created.";
    }
}

grouping multi-topology-config {
    description
        "Per-topology configuration.";
    container default-metric {
        uses default-metric-global-cfg;
        container level-1 {
            uses default-metric-global-cfg;
            description
                "Configuration specific to level 1.";
        }
        container level-2 {
            uses default-metric-global-cfg;
            description
                "Configuration specific to level 2.";
        }
        description
            "Default metric per-topology configuration.";
    }
    uses node-tag-config;
}

grouping interface-config {

```

```

description
    "Interface configuration grouping.";
uses admin-control;
leaf level-type {
    type level;
    default "level-all";
    description
        "IS-IS level of the interface.";
}
leaf lsp-pacing-interval {
    type rt-types:timer-value-milliseconds;
    units "milliseconds";
    default "33";
    description
        "Interval (in milliseconds) between LSP transmissions.";
}
leaf lsp-retransmit-interval {
    type rt-types:timer-value-seconds16;
    units "seconds";
    description
        "Interval (in seconds) between LSP retransmissions.";
}
leaf passive {
    type boolean;
    default "false";
    description
        "Indicates whether the interface is in passive mode (IS-IS
        is not running, but the network is advertised).";
}
leaf csnp-interval {
    type rt-types:timer-value-seconds16;
    units "seconds";
    default "10";
    description
        "Interval (in seconds) between Complete Sequence Number
        Packet (CSNP) messages.";
}
container hello-padding {
    leaf enabled {
        type boolean;
        default "true";
        description
            "IS-IS Hello padding activation.  Enabled by default.";
    }
    description
        "IS-IS Hello padding configuration.";
}
leaf mesh-group-enabled {
    type mesh-group-state;
    description
        "IS-IS interface mesh group state.";
}
leaf mesh-group {
    when "../mesh-group-enabled = 'mesh-set'" {
        description
            "Only valid when 'mesh-group-enabled' equals 'mesh-set'.";
    }
    type uint8;
    description
        "IS-IS interface mesh group ID.";
}
leaf interface-type {
    type interface-type;
    default "broadcast";
    description
        "Type of adjacency to be established for the interface.

```

```

        This dictates the type of Hello messages that are used.";
    }
    leaf-list tag {
        if-feature "prefix-tag";
        type uint32;
        description
            "List of tags associated with the interface.";
    }
    leaf-list tag64 {
        if-feature "prefix-tag64";
        type uint64;
        description
            "List of 64-bit tags associated with the interface.";
    }
    leaf node-flag {
        if-feature "node-flag";
        type boolean;
        default "false";
        description
            "Sets the prefix as a node representative prefix.";
    }
    container hello-authentication {
        uses hello-authentication-cfg;
        container level-1 {
            uses hello-authentication-cfg;
            description
                "Configuration specific to level 1.";
        }
        container level-2 {
            uses hello-authentication-cfg;
            description
                "Configuration specific to level 2.";
        }
        description
            "Authentication type to be used in Hello messages.";
    }
    container hello-interval {
        uses hello-interval-cfg-with-default;
        container level-1 {
            uses hello-interval-cfg;
            description
                "Configuration specific to level 1.";
        }
        container level-2 {
            uses hello-interval-cfg;
            description
                "Configuration specific to level 2.";
        }
        description
            "Interval between Hello messages.";
    }
    container hello-multiplier {
        uses hello-multiplier-cfg-with-default;
        container level-1 {
            uses hello-multiplier-cfg;
            description
                "Configuration specific to level 1.";
        }
        container level-2 {
            uses hello-multiplier-cfg;
            description
                "Configuration specific to level 2.";
        }
        description
            "Hello multiplier configuration.";
    }
}

```

```

container priority {
    must '../interface-type = "broadcast"' {
        error-message "Priority only applies to broadcast "
            + "interfaces.";
        description
            "Checks for a broadcast interface.";
    }
    uses priority-cfg-with-default;
    container level-1 {
        uses priority-cfg;
        description
            "Configuration specific to level 1.";
    }
    container level-2 {
        uses priority-cfg;
        description
            "Configuration specific to level 2.";
    }
    description
        "Priority for DIS election.";
}
container metric {
    uses metric-cfg-with-default;
    container level-1 {
        uses metric-cfg;
        description
            "Configuration specific to level 1.";
    }
    container level-2 {
        uses metric-cfg;
        description
            "Configuration specific to level 2.";
    }
    description
        "Metric configuration.";
}
container bfd {
    if-feature "bfd";
    description
        "BFD interface configuration.";
    uses bfd-types:client-cfg-parms;
    reference
        "RFC 5880: Bidirectional Forwarding Detection (BFD)
        RFC 5881: Bidirectional Forwarding Detection
        (BFD) for IPv4 and IPv6 (Single Hop)
        RFC 9314: YANG Data Model for Bidirectional Forwarding
        Detection (BFD)";
}
container address-families {
    if-feature "nlpid-control";
    list address-family-list {
        key "address-family";
        leaf address-family {
            type iana-rt-types:address-family;
            description
                "Address family.";
        }
    }
    description
        "List of address families.";
}
description
    "Interface address families.";
}
container mpls {
    container ldp {
        leaf igp-sync {

```

```

        if-feature "ldp-igp-sync";
        type boolean;
        default "false";
        description
            "Enables IGP/LDP synchronization.";
    }
    description
        "Configuration related to LDP.";
    }
    description
        "MPLS configuration for IS-IS interfaces.";
    }
    uses interface-fast-reroute-config;
}

grouping multi-topology-interface-config {
    description
        "IS-IS interface topology configuration.";
    container metric {
        uses metric-cfg;
        container level-1 {
            uses metric-cfg;
            description
                "Configuration specific to level 1.";
        }
        container level-2 {
            uses metric-cfg;
            description
                "Configuration specific to level 2.";
        }
    }
    description
        "Metric IS-IS interface configuration.";
    }
}

grouping interface-state {
    description
        "IS-IS interface operational state.";
    uses adjacency-state;
    uses event-counters;
    uses packet-counters;
    leaf discontinuity-time {
        type yang:date-and-time;
        description
            "The time of the most recent occasion at which any one
            or more of this IS-IS interface's counters suffered a
            discontinuity. If no such discontinuities have occurred
            since the IS-IS interface was last reinitialized, then
            this node contains the time the IS-IS interface was
            reinitialized, which normally occurs when it was
            created.";
    }
}

/* Grouping for the hostname database */

grouping hostname-db {
    container hostnames {
        config false;
        list hostname {
            key "system-id";
            leaf system-id {
                type system-id;
                description
                    "System ID associated with the hostname.";
            }
        }
    }
}

```

```

    leaf hostname {
        type string {
            length "1..255";
        }
        description
            "Hostname associated with the system ID
            as defined in RFC 5301.";
        reference
            "RFC 5301: Dynamic Hostname Exchange Mechanism
            for IS-IS";
    }
    description
        "List of system ID / hostname associations.";
}
description
    "Hostname-to-system-ID mapping database.";
}
description
    "Grouping for hostname-to-system-ID mapping database.";
}

/* Groupings for counters */

grouping system-counters {
    container system-counters {
        config false;
        list level {
            key "level";
            leaf level {
                type level-number;
                description
                    "IS-IS level.";
            }
        }
        leaf corrupted-lsps {
            type uint32;
            description
                "Number of corrupted in-memory LSPs detected.
                LSPs received from the wire with a bad
                checksum are silently dropped and not counted.
                LSPs received from the wire with parse errors
                are counted by 'lsp-errors'.";
        }
        leaf authentication-type-fails {
            type uint32;
            description
                "Number of authentication type mismatches.";
        }
        leaf authentication-fails {
            type uint32;
            description
                "Number of authentication key failures.";
        }
        leaf database-overload {
            type uint32;
            description
                "Number of times the database has become
                overloaded.";
        }
        leaf own-lsp-purge {
            type uint32;
            description
                "Number of times a zero-aged copy of the system's
                own LSP is received from some other IS-IS node.";
        }
        leaf manual-address-drop-from-area {
            type uint32;

```

```

        description
            "Number of times a manual address
            has been dropped from the area.";
    }
    leaf max-sequence {
        type uint32;
        description
            "Number of times the system has attempted
            to exceed the maximum sequence number.";
    }
    leaf sequence-number-skipped {
        type uint32;
        description
            "Number of times a sequence number skip has
            occurred.";
    }
    leaf id-len-mismatch {
        type uint32;
        description
            "Number of times a PDU is received with a
            different value for the ID field length
            than that of the receiving system.";
    }
    leaf partition-changes {
        type uint32;
        description
            "Number of partition changes detected.";
    }
    leaf lsp-errors {
        type uint32;
        description
            "Number of LSPs received with errors.";
    }
    leaf spf-runs {
        type uint32;
        description
            "Number of times SPF was run at this level.";
    }
    }
    description
        "List of supported levels.";
}
description
    "List of counters for the IS-IS protocol instance.";
}
description
    "Grouping for IS-IS system counters.";
}

grouping event-counters {
    container event-counters {
        config false;
        leaf adjacency-changes {
            type uint32;
            description
                "The number of times an adjacency state change has
                occurred on this interface.";
        }
        leaf adjacency-number {
            type uint32;
            description
                "The number of adjacencies on this interface.";
        }
    }
    leaf init-fails {
        type uint32;
        description
            "The number of times initialization of this interface has

```

```

        failed. This counts events such as Point-to-Point
        Protocol (PPP) Network Control Protocol (NCP) failures.
        Failures to form an adjacency are counted by
        'adjacency-rejects'.";
    }
    leaf adjacency-rejects {
        type uint32;
        description
            "The number of times an adjacency has been
            rejected on this interface.";
    }
    leaf id-len-mismatch {
        type uint32;
        description
            "The number of times an IS-IS PDU with an ID
            field length different from that for this
            system has been received on this interface.";
    }
    leaf max-area-addresses-mismatch {
        type uint32;
        description
            "The number of times an IS-IS PDU has been
            received on this interface with the
            max area address field differing from that of
            this system.";
    }
    leaf authentication-type-fails {
        type uint32;
        description
            "Number of authentication type mismatches.";
    }
    leaf authentication-fails {
        type uint32;
        description
            "Number of authentication key failures.";
    }
    leaf lan-dis-changes {
        type uint32;
        description
            "The number of times the DIS has changed on this
            interface at this level. If the interface type is
            'point-to-point', the count is zero.";
    }
    description
        "IS-IS interface event counters.";
}
description
    "Grouping for IS-IS interface event counters.";
}

grouping packet-counters {
    container packet-counters {
        config false;
        list level {
            key "level";
            leaf level {
                type level-number;
                description
                    "IS-IS level.";
            }
        }
        container iih {
            leaf in {
                type uint32;
                description
                    "Received IS-IS Hello (IIH) PDUs.";
            }
        }
    }
}

```

```

    leaf out {
        type uint32;
        description
            "Sent IIH PDUs.";
    }
    description
        "Number of IIH PDUs received/sent.";
}
container ish {
    leaf in {
        type uint32;
        description
            "Received Intermediate System Hello (ISH) PDUs.";
    }
    leaf out {
        type uint32;
        description
            "Sent ISH PDUs.";
    }
    description
        "ISH PDUs received/sent.";
}
container esh {
    leaf in {
        type uint32;
        description
            "Received End System Hello (ESH) PDUs.";
    }
    leaf out {
        type uint32;
        description
            "Sent ESH PDUs.";
    }
    description
        "Number of ESH PDUs received/sent.";
}
container lsp {
    leaf in {
        type uint32;
        description
            "Received Link State PDU (LSP) PDUs.";
    }
    leaf out {
        type uint32;
        description
            "Sent LSP PDUs.";
    }
    description
        "Number of LSP PDUs received/sent.";
}
container psnp {
    leaf in {
        type uint32;
        description
            "Received Partial Sequence Number PDU (PSNP) PDUs.";
    }
    leaf out {
        type uint32;
        description
            "Sent PSNP PDUs.";
    }
    description
        "Number of PSNP PDUs received/sent.";
}
container csnp {
    leaf in {

```

```

        type uint32;
        description
            "Received Complete Sequence Number PDU (CSNP) PDUs.";
    }
    leaf out {
        type uint32;
        description
            "Sent CSNP PDUs.";
    }
    description
        "Number of CSNP PDUs received/sent.";
}
container unknown {
    leaf in {
        type uint32;
        description
            "Received unknown PDUs.";
    }
    description
        "Number of unknown PDUs received.";
}
description
    "List of packet counters for supported levels.";
}
description
    "Packet counters per IS-IS level.";
}
description
    "Grouping for packet counters per IS-IS level.";
}

/* Groupings for various log buffers */

grouping spf-log {
    container spf-log {
        config false;
        list event {
            key "id";
            leaf id {
                type yang:counter32;
                description
                    "Event identifier.  A purely internal value.
                     The most recent events are expected to have a bigger
                     ID number.";
            }
        }
        leaf spf-type {
            type enumeration {
                enum full {
                    description
                        "Full SPF computation.";
                }
                enum route-only {
                    description
                        "SPF computation of route reachability
                         only.";
                }
            }
        }
        description
            "Type of SPF computation performed.";
    }
    leaf level {
        type level-number;
        description
            "IS-IS level number for the SPF computation.";
    }
    leaf schedule-timestamp {

```

```

        type yang:timestamp;
        description
            "Timestamp of when the SPF computation was
            scheduled.";
    }
    leaf start-timestamp {
        type yang:timestamp;
        description
            "Timestamp of when the SPF computation started.";
    }
    leaf end-timestamp {
        type yang:timestamp;
        description
            "Timestamp of when the SPF computation ended.";
    }
    list trigger-lsp {
        key "lsp";
        leaf lsp {
            type lsp-id;
            description
                "LSP ID of the LSP that triggered the SPF
                computation.";
        }
        leaf sequence {
            type uint32;
            description
                "Sequence number of the LSP that triggered the SPF
                computation.";
        }
        description
            "This list includes the LSPs that triggered the
            SPF computation.";
    }
    description
        "List of computation events.  Implemented as a
        wrapping buffer.";
}
description
    "This container lists the SPF computation events.";
}
description
    "Grouping for SPF log events.";
}

grouping lsp-log {
    container lsp-log {
        config false;
        list event {
            key "id";
            leaf id {
                type yang:counter32;
                description
                    "Event identifier.  A purely internal value.
                    The most recent events are expected to have a bigger
                    ID number.";
            }
        }
        leaf level {
            type level-number;
            description
                "IS-IS level number for the LSP.";
        }
    }
    container lsp {
        leaf lsp {
            type lsp-id;
            description
                "LSP ID of the LSP.";
        }
    }
}

```

```

    }
    leaf sequence {
        type uint32;
        description
            "Sequence number of the LSP.";
    }
    description
        "LSP identification container for either the received
        LSP or the locally generated LSP.";
}
leaf received-timestamp {
    type yang:timestamp;
    description
        "This is the timestamp when the LSP was received.
        In the case of a local LSP update, the timestamp refers
        to the LSP origination time.";
}
leaf reason {
    type identityref {
        base lsp-log-reason;
    }
    description
        "Type of LSP change.";
}
description
    "List of LSP events.  Implemented as a wrapping buffer.";
}
description
    "This container lists the LSP log.
    Local LSP modifications are also included in the list.";
}
description
    "Grouping for the LSP log.";
}

/* Groupings for the Link State Database (LSDB) descriptions */
/* Unknown TLV and sub-TLV descriptions */

grouping tlv {
    description
        "Type-Length-Value (TLV).";
    leaf type {
        type uint16;
        description
            "TLV type.";
    }
    leaf length {
        type uint16;
        description
            "TLV length (octets).";
    }
    leaf value {
        type yang:hex-string;
        description
            "TLV value.";
    }
}

grouping unknown-tlvs {
    description
        "Unknown TLVs grouping.  Used for unknown TLVs or
        unknown sub-TLVs.";
    container unknown-tlvs {
        description
            "All unknown TLVs.";
        list unknown-tlv {

```

```

        description
            "Unknown TLV.";
        uses tlv;
    }
}

/* TLVs and sub-TLVs for prefixes */

grouping prefix-reachability-attributes {
    description
        "Grouping for extended reachability attributes of an
        IPv4 or IPv6 prefix.";
    leaf external-prefix-flag {
        type boolean;
        description
            "External prefix flag.";
    }
    leaf readvertisement-flag {
        type boolean;
        description
            "Re-advertisement flag.";
    }
    leaf node-flag {
        type boolean;
        description
            "Node flag.";
    }
}

grouping prefix-ipv4-source-router-id {
    description
        "Grouping for the IPv4 source router ID of a prefix
        advertisement.";
    leaf ipv4-source-router-id {
        type inet:ipv4-address;
        description
            "IPv4 source router ID address.";
    }
}

grouping prefix-ipv6-source-router-id {
    description
        "Grouping for the IPv6 source router ID of a prefix
        advertisement.";
    leaf ipv6-source-router-id {
        type inet:ipv6-address;
        description
            "IPv6 source router ID address.";
    }
}

grouping prefix-attributes-extension {
    description
        "Prefix extended attributes as defined in RFC 7794.";
    reference
        "RFC 7794: IS-IS Prefix Attributes for Extended IPv4 and IPv6
        Reachability";
    uses prefix-reachability-attributes;
    uses prefix-ipv4-source-router-id;
    uses prefix-ipv6-source-router-id;
}

grouping prefix-ipv4-std {
    description
        "Grouping for attributes of an IPv4 standard prefix

```

```

    as defined in RFC 1195.";
reference
    "RFC 1195: Use of OSI IS-IS for routing in TCP/IP and
    dual environments";
leaf ip-prefix {
    type inet:ipv4-address;
    description
        "IPv4 prefix address.";
}
leaf prefix-len {
    type uint8;
    description
        "IPv4 prefix length (in bits).";
}
leaf i-e {
    type boolean;
    description
        "Internal or external (I/E) metric bit value.
        Set to 'false' to indicate an internal metric.";
}
container default-metric {
    leaf metric {
        type std-metric;
        description
            "Default IS-IS metric for the IPv4 prefix.";
    }
    description
        "IS-IS default metric container.";
}
container delay-metric {
    leaf metric {
        type std-metric;
        description
            "IS-IS delay metric for the IPv4 prefix.";
    }
    leaf supported {
        type boolean;
        default "false";
        description
            "Indicates whether the IS-IS delay metric is supported.";
    }
    description
        "IS-IS delay metric container.";
}
container expense-metric {
    leaf metric {
        type std-metric;
        description
            "IS-IS expense metric for the IPv4 prefix.";
    }
    leaf supported {
        type boolean;
        default "false";
        description
            "Indicates whether the IS-IS expense metric is supported.";
    }
    description
        "IS-IS expense metric container.";
}
container error-metric {
    leaf metric {
        type std-metric;
        description
            "This leaf describes the IS-IS error metric value.";
    }
    leaf supported {

```

```

        type boolean;
        default "false";
        description
            "Indicates whether the IS-IS error metric is supported.";
    }
    description
        "IS-IS error metric container.";
}
}

```

```

grouping prefix-ipv4-extended {
    description
        "Grouping for attributes of an IPv4 extended prefix
         as defined in RFC 5305.";
    reference
        "RFC 5305: IS-IS Extensions for Traffic Engineering";
    leaf up-down {
        type boolean;
        description
            "Value of the up/down bit.
             Set to 'true' when the prefix has been advertised down
             the hierarchy.";
    }
    leaf ip-prefix {
        type inet:ipv4-address;
        description
            "IPv4 prefix address.";
    }
    leaf prefix-len {
        type uint8;
        description
            "IPv4 prefix length (in bits).";
    }
    leaf metric {
        type wide-metric;
        description
            "IS-IS wide metric value.";
    }
    leaf-list tag {
        type uint32;
        description
            "List of 32-bit tags associated with the IPv4 prefix.";
    }
    leaf-list tag64 {
        type uint64;
        description
            "List of 64-bit tags associated with the IPv4 prefix.";
    }
    uses prefix-attributes-extension;
}

```

```

grouping prefix-ipv6-extended {
    description
        "Grouping for attributes of an IPv6 prefix
         as defined in RFC 5308.";
    reference
        "RFC 5308: Routing IPv6 with IS-IS";
    leaf up-down {
        type boolean;
        description
            "Value of the up/down bit.
             Set to 'true' when the prefix has been advertised down
             the hierarchy.";
    }
    leaf ip-prefix {
        type inet:ipv6-address;
    }
}

```

```

        description
            "IPv6 prefix address.";
    }
    leaf prefix-len {
        type uint8;
        description
            "IPv6 prefix length (in bits).";
    }
    leaf metric {
        type wide-metric;
        description
            "IS-IS wide metric value.";
    }
    leaf-list tag {
        type uint32;
        description
            "List of 32-bit tags associated with the IPv6 prefix.";
    }
    leaf-list tag64 {
        type uint64;
        description
            "List of 64-bit tags associated with the IPv6 prefix.";
    }
    uses prefix-attributes-extension;
}

/* TLVs and sub-TLVs for neighbors */

grouping neighbor-link-attributes {
    description
        "Grouping for link attributes as defined
        in RFC 5029.";
    reference
        "RFC 5029: Definition of an IS-IS Link Attribute Sub-TLV";
    leaf link-attributes-flags {
        type uint16;
        description
            "Flags for the link attributes.";
    }
}

grouping neighbor-gmpls-extensions {
    description
        "Grouping for GMPLS attributes of a neighbor as defined
        in RFC 5307.";
    reference
        "RFC 5307: IS-IS Extensions in Support of Generalized
        Multi-Protocol Label Switching (GMPLS)";
    leaf link-local-id {
        type uint32;
        description
            "Local identifier of the link.";
    }
    leaf remote-local-id {
        type uint32;
        description
            "Remote identifier of the link.";
    }
    leaf protection-capability {
        type uint8;
        description
            "Describes the protection capabilities
            of the link. This is the value of the
            first octet of the sub-TLV type 20 value.";
    }
    container interface-switching-capability {

```

```

description
    "Interface switching capabilities of the link.";
leaf switching-capability {
    type uint8;
    description
        "Switching capability of the link.";
}
leaf encoding {
    type uint8;
    description
        "Type of encoding of the LSP being used.";
}
container max-lsp-bandwidths {
    description
        "Per-priority maximum LSP bandwidths.";
    list max-lsp-bandwidth {
        leaf priority {
            type uint8 {
                range "0 .. 7";
            }
            description
                "Priority from 0 to 7.";
        }
        leaf bandwidth {
            type rt-types:bandwidth-ieee-float32;
            description
                "Maximum LSP bandwidth.";
        }
        description
            "List of maximum LSP bandwidths for different
            priorities.";
    }
}
container tdm-specific {
    when '../switching-capability = 100';
    description
        "Switching-capability-specific information applicable
        when the switching type is Time-Division Multiplexing
        (TDM).";
    leaf minimum-lsp-bandwidth {
        type rt-types:bandwidth-ieee-float32;
        description
            "Minimum LSP bandwidth.";
    }
    leaf indication {
        type uint8;
        description
            "Indicates whether the interface supports Standard
            or Arbitrary SONET/SDH (Synchronous Optical Network /
            Synchronous Digital Hierarchy).";
    }
}
container psc-specific {
    when "../switching-capability >= 1 and
        ../switching-capability <= 4";
    description
        "Switching-capability-specific information applicable
        when the switching type is PSC1, PSC2, PSC3, or PSC4
        ('PSC' stands for 'Packet Switching Capability').";
    leaf minimum-lsp-bandwidth {
        type rt-types:bandwidth-ieee-float32;
        description
            "Minimum LSP bandwidth.";
    }
    leaf mtu {
        type uint16;
    }
}

```

```

        units "bytes";
        description
            "Interface MTU.";
    }
}
}

grouping neighbor-extended-te-extensions {
    description
        "Grouping for TE attributes of a neighbor as defined
        in RFC 8570.";
    reference
        "RFC 8570: IS-IS Traffic Engineering (TE) Metric Extensions";
    container unidirectional-link-delay {
        description
            "Container for the average delay
            from the local neighbor to the remote neighbor.";
        container flags {
            leaf-list unidirectional-link-delay-subtlv-flags {
                type identityref {
                    base unidirectional-link-delay-subtlv-flag;
                }
                description
                    "This list contains identities for the bits that
                    are set.";
            }
            description
                "Unidirectional Link Delay sub-TLV flags.";
        }
        leaf value {
            type uint32;
            units "usec";
            description
                "Delay value expressed in microseconds.";
        }
    }
}
container min-max-unidirectional-link-delay {
    description
        "Container for the minimum and maximum delay
        from the local neighbor to the remote neighbor.";
    container flags {
        leaf-list min-max-unidirectional-link-delay-subtlv-flags {
            type identityref {
                base min-max-unidirectional-link-delay-subtlv-flag;
            }
            description
                "This list contains identities for the bits that
                are set.";
        }
        description
            "Min/Max Unidirectional Link Delay sub-TLV flags.";
    }
    leaf min-value {
        type uint32;
        units "usec";
        description
            "Minimum delay value expressed in microseconds.";
    }
    leaf max-value {
        type uint32;
        units "usec";
        description
            "Maximum delay value expressed in microseconds.";
    }
}
}

```

```

container unidirectional-link-delay-variation {
  description
    "Container for the average delay variation
    from the local neighbor to the remote neighbor.";
  leaf value {
    type uint32;
    units "usec";
    description
      "Delay variation value expressed in microseconds.";
  }
}
container unidirectional-link-loss {
  description
    "Container for packet loss from the local neighbor to the
    remote neighbor.";
  container flags {
    leaf-list unidirectional-link-loss-subtlv-flags {
      type identityref {
        base unidirectional-link-loss-subtlv-flag;
      }
      description
        "This list contains identities for the bits that
        are set.";
    }
    description
      "Unidirectional Link Loss sub-TLV flags.";
  }
  leaf value {
    type uint32;
    units "percent";
    description
      "Link packet loss expressed as a percentage of
      the total traffic sent over a configurable interval.";
  }
}
container unidirectional-link-residual-bandwidth {
  description
    "Container for the residual bandwidth
    from the local neighbor to the remote neighbor.";
  leaf value {
    type rt-types:bandwidth-ieee-float32;
    units "Bps";
    description
      "Residual bandwidth.";
  }
}
container unidirectional-link-available-bandwidth {
  description
    "Container for the available bandwidth
    from the local neighbor to the remote neighbor.";
  leaf value {
    type rt-types:bandwidth-ieee-float32;
    units "Bps";
    description
      "Available bandwidth.";
  }
}
container unidirectional-link-utilized-bandwidth {
  description
    "Container for the utilized bandwidth
    from the local neighbor to the remote neighbor.";
  leaf value {
    type rt-types:bandwidth-ieee-float32;
    units "Bps";
    description
      "Utilized bandwidth.";
  }
}

```

```

    }
  }
}

grouping neighbor-te-extensions {
  description
    "Grouping for TE attributes of a neighbor as defined
    in RFC 5305.";
  reference
    "RFC 5305: IS-IS Extensions for Traffic Engineering";
  leaf admin-group {
    type uint32;
    description
      "Administrative Group / Resource Class/Color.";
  }
  container local-if-ipv4-addrs {
    description
      "All local interface IPv4 addresses.";
    leaf-list local-if-ipv4-addr {
      type inet:ipv4-address;
      description
        "List of local interface IPv4 addresses.";
    }
  }
  container remote-if-ipv4-addrs {
    description
      "All remote interface IPv4 addresses.";
    leaf-list remote-if-ipv4-addr {
      type inet:ipv4-address;
      description
        "List of remote interface IPv4 addresses.";
    }
  }
  leaf te-metric {
    type uint32;
    description
      "TE metric.";
  }
  leaf max-bandwidth {
    type rt-types:bandwidth-ieee-float32;
    description
      "Maximum bandwidth.";
  }
  leaf max-reservable-bandwidth {
    type rt-types:bandwidth-ieee-float32;
    description
      "Maximum reservable bandwidth.";
  }
  container unreserved-bandwidths {
    description
      "All unreserved bandwidths.";
    list unreserved-bandwidth {
      leaf priority {
        type uint8 {
          range "0 .. 7";
        }
        description
          "Priority from 0 to 7.";
      }
      leaf unreserved-bandwidth {
        type rt-types:bandwidth-ieee-float32;
        description
          "Unreserved bandwidth.";
      }
    }
    description
      "List of unreserved bandwidths for different

```

```

        priorities.";
    }
}

grouping neighbor-extended {
    description
        "Grouping for attributes of an IS-IS extended neighbor.";
    leaf neighbor-id {
        type extended-system-id;
        description
            "System ID of the extended neighbor.";
    }
    container instances {
        description
            "List of all adjacencies between the local
            system and the neighbor system ID.";
        list instance {
            key "id";
            leaf id {
                type uint32;
                description
                    "Unique identifier of an instance of a
                    particular neighbor.";
            }
            leaf metric {
                type wide-metric;
                description
                    "IS-IS wide metric for the extended neighbor.";
            }
            uses neighbor-gmpls-extensions;
            uses neighbor-te-extensions;
            uses neighbor-extended-te-extensions;
            uses neighbor-link-attributes;
            uses unknown-tlvs;
            description
                "Instance of a particular adjacency.";
        }
    }
}

grouping neighbor {
    description
        "IS-IS standard neighbor grouping.";
    leaf neighbor-id {
        type extended-system-id;
        description
            "IS-IS neighbor system ID.";
    }
    container instances {
        description
            "List of all adjacencies between the local
            system and the neighbor system ID.";
        list instance {
            key "id";
            leaf id {
                type uint32;
                description
                    "Unique identifier of an instance of a
                    particular neighbor.";
            }
            leaf i-e {
                type boolean;
                description
                    "Internal or external (I/E) metric bit value.
                    Set to 'false' to indicate an internal metric.";
            }
        }
    }
}

```

```

    }
    container default-metric {
        leaf metric {
            type std-metric;
            description
                "IS-IS default metric value.";
        }
        description
            "IS-IS default metric container.";
    }
    container delay-metric {
        leaf metric {
            type std-metric;
            description
                "IS-IS delay metric value.";
        }
        leaf supported {
            type boolean;
            default "false";
            description
                "IS-IS delay metric supported.";
        }
        description
            "IS-IS delay metric container.";
    }
    container expense-metric {
        leaf metric {
            type std-metric;
            description
                "IS-IS expense metric value.";
        }
        leaf supported {
            type boolean;
            default "false";
            description
                "IS-IS expense metric supported.";
        }
        description
            "IS-IS expense metric container.";
    }
    container error-metric {
        leaf metric {
            type std-metric;
            description
                "IS-IS error metric value.";
        }
        leaf supported {
            type boolean;
            default "false";
            description
                "IS-IS error metric supported.";
        }
        description
            "IS-IS error metric container.";
    }
    description
        "Instance of a particular adjacency as defined in
        ISO 10589.";
    reference
        "ISO 10589: Intermediate System to Intermediate System
        intra-domain routing information exchange protocol
        for use in conjunction with the protocol for providing
        the connectionless-mode network service (ISO 8473)";
}
}
}

```

```

/* Top-level TLVs */

grouping tlv132-ipv4-addresses {
    leaf-list ipv4-addresses {
        type inet:ipv4-address;
        description
            "List of IPv4 addresses of the IS-IS node.  The IS-IS
            reference is TLV 132.";
    }
    description
        "Grouping for TLV 132.";
}

grouping tlv232-ipv6-addresses {
    leaf-list ipv6-addresses {
        type inet:ipv6-address;
        description
            "List of IPv6 addresses of the IS-IS node.  The IS-IS
            reference is TLV 232.";
    }
    description
        "Grouping for TLV 232.";
}

grouping tlv134-ipv4-te-rid {
    leaf ipv4-te-routerid {
        type inet:ipv4-address;
        description
            "IPv4 Traffic Engineering router ID of the IS-IS node.
            The IS-IS reference is TLV 134.";
    }
    description
        "Grouping for TLV 134.";
}

grouping tlv140-ipv6-te-rid {
    leaf ipv6-te-routerid {
        type inet:ipv6-address;
        description
            "IPv6 Traffic Engineering router ID of the IS-IS node.
            The IS-IS reference is TLV 140.";
    }
    description
        "Grouping for TLV 140.";
}

grouping tlv129-protocols {
    leaf-list protocol-supported {
        type uint8;
        description
            "List of supported protocols of the IS-IS node.
            The IS-IS reference is TLV 129.";
    }
    description
        "Grouping for TLV 129.";
}

grouping tlv137-hostname {
    leaf dynamic-hostname {
        type string;
        description
            "Hostname of the IS-IS node.  The IS-IS reference
            is TLV 137.";
    }
    description

```

```

    "Grouping for TLV 137.";
}

grouping tlv10-authentication {
    container authentication {
        leaf authentication-type {
            type identityref {
                base key-chain:crypto-algorithm;
            }
            description
                "Authentication type to be used with an IS-IS node.";
        }
        leaf authentication-key {
            type string;
            description
                "Authentication key to be used. For security reasons,
                the authentication key MUST NOT be presented in
                a cleartext format in response to any request
                (e.g., via get or get-config).";
        }
        description
            "IS-IS node authentication information container. The
            IS-IS reference is TLV 10.";
    }
    description
        "Grouping for TLV 10.";
}

grouping tlv229-mt {
    container mt-entries {
        list topology {
            description
                "List of topologies supported.";
            leaf mt-id {
                type uint16 {
                    range "0 .. 4095";
                }
                description
                    "Multi-Topology (MT) identifier of the topology.";
            }
            container attributes {
                leaf-list flags {
                    type identityref {
                        base tlv229-flag;
                    }
                    description
                        "This list contains identities for the bits that
                        are set.";
                }
                description
                    "TLV 229 flags.";
            }
        }
        description
            "IS-IS node topology information container. The
            IS-IS reference is TLV 229.";
    }
    description
        "Grouping for TLV 229.";
}

grouping tlv242-router-capabilities {
    container router-capabilities {
        list router-capability {
            container flags {
                leaf-list router-capability-flags {

```

```

        type identityref {
            base router-capability-flag;
        }
        description
            "This list contains identities for the bits that
            are set.";
    }
    description
        "Router Capability flags.";
}
container node-tags {
    if-feature "node-tag";
    list node-tag {
        leaf tag {
            type uint32;
            description
                "Node tag value.";
        }
        description
            "List of tags.";
    }
    description
        "Container for node administrative tags.";
}
uses unknown-tlvs;
description
    "IS-IS node capabilities. This list element may
    be extended with detailed information. The IS-IS
    reference is TLV 242.";
}
description
    "List of Router Capability TLVs.";
}
description
    "Grouping for TLV 242.";
}

grouping tlv138-srlg {
    description
        "Grouping for TLV 138.";
    container links-srlgs {
        list links {
            leaf neighbor-id {
                type extended-system-id;
                description
                    "System ID of the extended neighbor.";
            }
            leaf flags {
                type uint8;
                description
                    "Flags associated with the link.";
            }
            leaf link-local-id {
                type union {
                    type inet:ip-address;
                    type uint32;
                }
                description
                    "Local identifier of the link.
                    It could be an IPv4 address or a local identifier.";
            }
            leaf link-remote-id {
                type union {
                    type inet:ip-address;
                    type uint32;
                }
            }
        }
    }
}

```

```

        description
            "Remote identifier of the link.
            It could be an IPv4 address or a remotely learned
            identifier.";
    }
    container srlgs {
        description
            "List of SRLGs.";
        leaf-list srlg {
            type uint32;
            description
                "SRLG value of the link.";
        }
    }
    description
        "SRLG attribute of a link.";
}
description
    "List of links with SRLGs.";
}
}

/* Grouping for LSDB descriptions */

grouping lsp-entry {
    description
        "IS-IS LSP database entry grouping.";
    leaf decoded-completed {
        type boolean;
        description
            "The IS-IS LSP body has been fully decoded.";
    }
    leaf raw-data {
        type yang:hex-string;
        description
            "The hexadecimal representation of the complete LSP
            as received or originated, in network byte order.";
    }
    leaf lsp-id {
        type lsp-id;
        description
            "LSP ID of the LSP.";
    }
    leaf checksum {
        type uint16;
        description
            "LSP checksum.";
    }
    leaf remaining-lifetime {
        type uint16;
        units "seconds";
        description
            "Remaining lifetime (in seconds) until LSP expiration.";
    }
    leaf sequence {
        type uint32;
        description
            "This leaf describes the sequence number of the LSP.";
    }
    container attributes {
        leaf-list lsp-flags {
            type identityref {
                base lsp-flag;
            }
        }
        description
            "This list contains identities for the bits that

```

```

        are set.";
    }
    description
        "LSP attributes.";
}
uses tlv132-ipv4-addresses;
uses tlv232-ipv6-addresses;
uses tlv134-ipv4-te-rid;
uses tlv140-ipv6-te-rid;
uses tlv129-protocols;
uses tlv137-hostname;
uses tlv10-authentication;
uses tlv229-mt;
uses tlv242-router-capabilities;
uses tlv138-srlg;
uses unknown-tlvs;
container is-neighbor {
    list neighbor {
        key "neighbor-id";
        uses neighbor;
        description
            "List of neighbors.";
    }
    description
        "Standard IS neighbors container. The IS-IS reference is
        TLV 2.";
}
container extended-is-neighbor {
    list neighbor {
        key "neighbor-id";
        uses neighbor-extended;
        description
            "List of extended IS neighbors.";
    }
    description
        "Standard IS extended neighbors container. The IS-IS
        reference is TLV 22.";
}
container ipv4-internal-reachability {
    list prefixes {
        uses prefix-ipv4-std;
        description
            "List of prefixes.";
    }
    description
        "IPv4 internal reachability information container.
        The IS-IS reference is TLV 128.";
}
container ipv4-external-reachability {
    list prefixes {
        uses prefix-ipv4-std;
        description
            "List of prefixes.";
    }
    description
        "IPv4 external reachability information container. The
        IS-IS reference is TLV 130.";
}
container extended-ipv4-reachability {
    list prefixes {
        uses prefix-ipv4-extended;
        uses unknown-tlvs;
        description
            "List of prefixes.";
    }
    description

```

```

        "IPv4 extended reachability information container.  The
        IS-IS reference is TLV 135.";
    }
    container mt-is-neighbor {
        list neighbor {
            leaf mt-id {
                type uint16 {
                    range "0 .. 4095";
                }
                description
                    "Multi-Topology (MT) identifier.";
            }
            uses neighbor-extended;
            description
                "List of neighbors.";
        }
        description
            "IS-IS MT neighbor container.  The IS-IS reference is
            TLV 223.";
    }
    container mt-extended-ipv4-reachability {
        list prefixes {
            leaf mt-id {
                type uint16 {
                    range "0 .. 4095";
                }
                description
                    "MT identifier.";
            }
            uses prefix-ipv4-extended;
            uses unknown-tlvs;
            description
                "List of extended prefixes.";
        }
        description
            "IPv4 MT extended reachability information container.
            The IS-IS reference is TLV 235.";
        reference
            "RFC 5120: M-ISIS: Multi Topology (MT) Routing in
            Intermediate System to Intermediate Systems (IS-ISs)";
    }
    container mt-ipv6-reachability {
        list prefixes {
            leaf mt-id {
                type uint16 {
                    range "0 .. 4095";
                }
                description
                    "MT identifier.";
            }
            uses prefix-ipv6-extended;
            uses unknown-tlvs;
            description
                "List of IPv6 extended prefixes.";
        }
        description
            "IPv6 MT extended reachability information container.
            The IS-IS reference is TLV 237.";
        reference
            "RFC 5120: M-ISIS: Multi Topology (MT) Routing in
            Intermediate System to Intermediate Systems (IS-ISs)";
    }
    container ipv6-reachability {
        list prefixes {
            uses prefix-ipv6-extended;
            uses unknown-tlvs;
        }
    }

```

```

        description
            "List of IPv6 prefixes.";
    }
    description
        "IPv6 reachability information container.  The IS-IS
        reference is TLV 236.";
    }
}

grouping lsdb {
    description
        "Link State Database (LSDB) grouping.";
    container database {
        config false;
        list levels {
            key "level";
            leaf level {
                type level-number;
                description
                    "LSDB level number (1 or 2).";
            }
            list lsp {
                key "lsp-id";
                uses lsp-entry;
                description
                    "List of LSPs in the LSDB.";
            }
            description
                "List of LSPs for the LSDB-level container.";
        }
        description
            "IS-IS LSDB container.";
    }
}

/* Augmentations */

augment "/rt:routing/"
    + "rt:ribs/rt:rib/rt:routes/rt:route" {
    when "derived-from-or-self(rt:source-protocol, 'isis:isis')" {
        description
            "IS-IS-specific route attributes.";
    }
    uses route-content;
    description
        "This augments the route object in the Routing Information
        Base (RIB) with IS-IS-specific attributes.";
}

augment "/if:interfaces/if:interface" {
    leaf clns-mtu {
        if-feature "osi-interface";
        type uint16;
        description
            "Connectionless-mode Network Service (CLNS) MTU of the
            interface.";
    }
    description
        "ISO-specific interface parameters.";
}

augment "/rt:routing/rt:control-plane-protocols/"
    + "rt:control-plane-protocol" {
    when "derived-from-or-self(rt:type, 'isis:isis')" {
        description
            "This augmentation is only valid when the routing protocol

```

```

        instance type is 'isis'.";
    }
    description
        "This augments a routing protocol instance with IS-IS-specific
        parameters.";
    container isis {
        must 'count(area-address) > 0' {
            error-message "At least one area address must be "
                + "configured.";
            description
                "Enforces the configuration of at least one area.";
        }
        uses instance-config;
        uses instance-state;
        container topologies {
            if-feature "multi-topology";
            list topology {
                key "name";
                leaf enabled {
                    type boolean;
                    description
                        "Enables the topology configuration.";
                }
                leaf name {
                    type leafref {
                        path "../../../../../rt:ribs/rt:rib/rt:name";
                    }
                    description
                        "RIB corresponding to the topology.";
                }
                uses multi-topology-config;
                description
                    "List of topologies.";
            }
            description
                "MT container.";
        }
        container interfaces {
            list interface {
                key "name";
                leaf name {
                    type if:interface-ref;
                    description
                        "Reference to the interface within
                        the routing instance.";
                }
            }
            uses interface-config;
            uses interface-state;
            container topologies {
                if-feature "multi-topology";
                list topology {
                    key "name";
                    leaf name {
                        type leafref {
                            path "../../../../../rt:ribs/rt:rib/rt:name";
                        }
                    }
                    description
                        "RIB corresponding to the topology.";
                }
                uses multi-topology-interface-config;
                description
                    "List of interface topologies.";
            }
            description
                "MT container.";
        }
    }
}

```

```

    }
    description
        "List of IS-IS interfaces.";
    }
    description
        "Configuration container specific to IS-IS interfaces.";
    }
    description
        "IS-IS configuration/state top-level container.";
}
}

/* RPC methods */

rpc clear-adjacency {
    description
        "This RPC request clears a particular set of IS-IS
        adjacencies.  If the operation fails for an internal
        reason, then the 'error-tag' and 'error-app-tag' should be
        set indicating the reason for the failure.";
    reference
        "RFC 6241: Network Configuration Protocol (NETCONF)";
    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 IS-IS protocol instance whose IS-IS
                adjacency is being cleared.

                If the corresponding IS-IS 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'.";
        }
        leaf level {
            type level;
            description
                "IS-IS level of the adjacency to be cleared.  If the
                IS-IS level is 'level-all', level-1 and level-2
                adjacencies would both be cleared.

                If the value provided is different from the value
                authorized in the enum type, then the operation
                SHALL fail with an 'error-tag' of 'data-missing' and
                an 'error-app-tag' of 'bad-isis-level'.";
        }
        leaf interface {
            type if:interface-ref;
            description
                "IS-IS interface name.

                If the corresponding IS-IS interface doesn't exist,
                then the operation SHALL fail with an 'error-tag' of
                'data-missing' and an 'error-app-tag' of
                'isis-interface-not-found'.";
        }
    }
}

rpc clear-database {
    description
        "This RPC request clears a particular IS-IS database."

```

Additionally, all neighbor adjacencies will be forced to the DOWN state and self-originated LSPs will be reoriginated. If the operation fails for an IS-IS internal reason, then the 'error-tag' and 'error-app-tag' should be set indicating the reason for the failure.";

```
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 IS-IS protocol instance whose IS-IS
        database or databases are being cleared.

        If the corresponding IS-IS 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'.";
  }
  leaf level {
    type level;
    description
      "IS-IS level of the adjacency to be cleared. If the
        IS-IS level is 'level-all', the databases for both
        level 1 and level 2 would be cleared.

        If the value provided is different from the value
        authorized in the enum type, then the operation
        SHALL fail with an 'error-tag' of 'data-missing' and
        an 'error-app-tag' of 'bad-isis-level'.";
  }
}
}

/* Notifications */

notification database-overload {
  uses notification-instance-hdr;
  leaf overload {
    type enumeration {
      enum off {
        description
          "Indicates that the IS-IS instance has left the
            overload state.";
      }
      enum on {
        description
          "Indicates that the IS-IS instance has entered the
            overload state.";
      }
    }
    description
      "New overload state of the IS-IS instance.";
  }
  description
    "This notification is sent when an IS-IS instance
      overload state changes.";
}

notification lsp-too-large {
  uses notification-instance-hdr;
  uses notification-interface-hdr;
  leaf pdu-size {
    type uint32;
```

```

        description
            "Size of the LSP PDU.";
    }
    leaf lsp-id {
        type lsp-id;
        description
            "LSP ID.";
    }
    description
        "This notification is sent when an attempt to propagate
        an LSP that is larger than the dataLinkBlockSize (ISO 10589)
        for the circuit occurs. The generation of the notification
        must be throttled with at least 5 seconds between successive
        notifications.";
    reference
        "ISO 10589: Intermediate System to Intermediate System
        intra-domain routing information exchange protocol
        for use in conjunction with the protocol for providing
        the connectionless-mode network service (ISO 8473)";
}

notification if-state-change {
    uses notification-instance-hdr;
    uses notification-interface-hdr;
    leaf state {
        type if-state-type;
        description
            "Interface state.";
    }
    description
        "This notification is sent when an interface
        state change is detected.";
}

notification corrupted-lsp-detected {
    uses notification-instance-hdr;
    leaf lsp-id {
        type lsp-id;
        description
            "LSP ID.";
    }
    description
        "This notification is sent when an LSP that was stored in
        memory has become corrupted.";
}

notification attempt-to-exceed-max-sequence {
    uses notification-instance-hdr;
    leaf lsp-id {
        type lsp-id;
        description
            "LSP ID.";
    }
    description
        "This notification is sent when the system
        wraps the 32-bit sequence counter of an LSP.";
}

notification id-len-mismatch {
    uses notification-instance-hdr;
    uses notification-interface-hdr;
    leaf pdu-field-len {
        type uint8;
        description
            "Value for the system ID length in the received PDU.";
    }
}

```

```

leaf raw-pdu {
    type binary;
    description
        "Received raw PDU.";
}
description
    "This notification is sent when a PDU with a different value
    for the system ID length is received. The generation of the
    notification must be throttled with at least 5 seconds
    between successive notifications.";
}

notification max-area-addresses-mismatch {
    uses notification-instance-hdr;
    uses notification-interface-hdr;
    leaf max-area-addresses {
        type uint8;
        description
            "Received number of supported areas.";
    }
    leaf raw-pdu {
        type binary;
        description
            "Received raw PDU.";
    }
    description
        "This notification is sent when a PDU with a different value
        for the Maximum Area Addresses has been received. The
        generation of the notification must be throttled with
        at least 5 seconds between successive notifications.";
}

notification own-lsp-purge {
    uses notification-instance-hdr;
    uses notification-interface-hdr;
    leaf lsp-id {
        type lsp-id;
        description
            "LSP ID.";
    }
    description
        "This notification is sent when the system receives
        a PDU with its own system ID and zero age.";
}

notification sequence-number-skipped {
    uses notification-instance-hdr;
    uses notification-interface-hdr;
    leaf lsp-id {
        type lsp-id;
        description
            "LSP ID.";
    }
    description
        "This notification is sent when the system receives a
        PDU with its own system ID and different contents. The
        system has to originate the LSP with a higher sequence
        number.";
}

notification authentication-type-failure {
    uses notification-instance-hdr;
    uses notification-interface-hdr;
    leaf raw-pdu {
        type binary;
        description

```

```

        "Received raw PDU.";
    }
    description
        "This notification is sent when the system receives a
        PDU with the wrong authentication type field.
        The generation of the notification must be throttled
        with at least 5 seconds between successive notifications.";
}

notification authentication-failure {
    uses notification-instance-hdr;
    uses notification-interface-hdr;
    leaf raw-pdu {
        type binary;
        description
            "Received raw PDU.";
    }
    description
        "This notification is sent when the system receives
        a PDU on which authentication fails. The generation of the
        notification must be throttled with at least 5 seconds
        between successive notifications.";
}

notification version-skew {
    uses notification-instance-hdr;
    uses notification-interface-hdr;
    leaf protocol-version {
        type uint8;
        description
            "Protocol version received in the PDU.";
    }
    leaf raw-pdu {
        type binary;
        description
            "Received raw PDU.";
    }
    description
        "This notification is sent when the system receives a
        PDU with a different protocol version number.
        The generation of the notification must be throttled
        with at least 5 seconds between successive notifications.";
}

notification area-mismatch {
    uses notification-instance-hdr;
    uses notification-interface-hdr;
    leaf raw-pdu {
        type binary;
        description
            "Received raw PDU.";
    }
    description
        "This notification is sent when the system receives a
        Hello PDU from an IS that does not share any area
        address. The generation of the notification must be
        throttled with at least 5 seconds between successive
        notifications.";
}

notification rejected-adjacency {
    uses notification-instance-hdr;
    uses notification-interface-hdr;
    leaf raw-pdu {
        type binary;
        description

```

```

        "Received raw PDU.";
    }
    leaf reason {
        type string {
            length "0..255";
        }
        description
            "The system may provide a reason to reject the
            adjacency. If the reason is not available,
            the reason string will not be returned.
            The expected format is a single line of text.";
    }
    description
        "This notification is sent when the system receives a
        Hello PDU from an IS but does not establish an adjacency
        for some reason. The generation of the notification
        must be throttled with at least 5 seconds between
        successive notifications.";
}

notification protocols-supported-mismatch {
    uses notification-instance-hdr;
    uses notification-interface-hdr;
    leaf raw-pdu {
        type binary;
        description
            "Received raw PDU.";
    }
    leaf-list protocols {
        type uint8;
        description
            "List of protocols supported by the remote system.";
    }
    description
        "This notification is sent when the system receives a
        non-pseudonode LSP that has no matching protocols
        supported. The generation of the notification must be
        throttled with at least 5 seconds between successive
        notifications.";
}

notification lsp-error-detected {
    uses notification-instance-hdr;
    uses notification-interface-hdr;
    leaf lsp-id {
        type lsp-id;
        description
            "LSP ID.";
    }
    leaf raw-pdu {
        type binary;
        description
            "Received raw PDU.";
    }
    leaf error-offset {
        type uint32;
        description
            "If the problem is a malformed TLV, the error offset
            points to the start of the TLV. If the problem is with
            the LSP header, the error offset points to the errant
            byte.";
    }
    leaf tlv-type {
        type uint8;
        description
            "If the problem is a malformed TLV, the TLV type is set

```

```

        to the type value of the suspicious TLV. Otherwise,
        this leaf is not present.";
    }
    description
        "This notification is sent when the system receives an
        LSP with a parse error. The generation of the notification
        must be throttled with at least 5 seconds between
        successive notifications.";
}

notification adjacency-state-change {
    uses notification-instance-hdr;
    uses notification-interface-hdr;
    leaf neighbor {
        type string {
            length "1..255";
        }
        description
            "Name of the neighbor. It corresponds to the hostname
            associated with the system ID of the neighbor in the
            mapping database (RFC 5301). If the name of the neighbor
            is not available, it is not returned.";
        reference
            "RFC 5301: Dynamic Hostname Exchange Mechanism for IS-IS";
    }
    leaf neighbor-system-id {
        type system-id;
        description
            "Neighbor system ID.";
    }
    leaf state {
        type adj-state-type;
        description
            "New state of the IS-IS adjacency.";
    }
    leaf reason {
        type string {
            length "1..255";
        }
        description
            "If the adjacency is going to the 'down' state, this leaf
            provides a reason for the adjacency going down. The reason
            is provided as text. If the adjacency is going to the 'up'
            state, no reason is provided. The expected format is a
            single line of text.";
    }
    description
        "This notification is sent when an IS-IS adjacency
        moves to the 'up' state or the 'down' state.";
}

notification lsp-received {
    uses notification-instance-hdr;
    uses notification-interface-hdr;
    leaf lsp-id {
        type lsp-id;
        description
            "LSP ID.";
    }
    leaf sequence {
        type uint32;
        description
            "Sequence number of the received LSP.";
    }
    leaf received-timestamp {
        type yang:timestamp;
    }
}

```

```

        description
            "Timestamp when the LSP was received.";
    }
    leaf neighbor-system-id {
        type system-id;
        description
            "Neighbor system ID of the LSP sender.";
    }
    description
        "This notification is sent when an LSP is received.
        The generation of the notification must be throttled with
        at least 5 seconds between successive notifications.";
}

notification lsp-generation {
    uses notification-instance-hdr;
    leaf lsp-id {
        type lsp-id;
        description
            "LSP ID.";
    }
    leaf sequence {
        type uint32;
        description
            "Sequence number of the received LSP.";
    }
    leaf send-timestamp {
        type yang:timestamp;
        description
            "Timestamp when the LSP was regenerated.";
    }
    description
        "This notification is sent when an LSP is regenerated.
        The generation of the notification must be throttled with
        at least 5 seconds between successive notifications.";
}
}
<CODE ENDS>

```

## 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 this YANG module 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:

```

/isis

/isis/interfaces/interface[name]

```

For IS-IS, the ability to modify IS-IS configuration will allow the entire IS-IS domain to be compromised, including forming adjacencies with unauthorized routers to misroute traffic or mount a massive Denial-of-Service (DoS) attack. For example, adding IS-IS on any unprotected interface could allow an IS-IS adjacency to be formed with an unauthorized and malicious neighbor. Once an adjacency is formed, traffic could be hijacked. As a simpler example, a DoS attack could be mounted by changing the cost of an IS-IS interface to be asymmetric, such that a hard routing loop ensues. In general, unauthorized modification of most IS-IS features will pose its own set of security risks; therefore, the Security Considerations sections in the respective reference RFCs should be consulted.

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:

/isis/database

/isis/local-rib

Exposure of the Link State Database (LSDB) will reveal the detailed topology of the network. Similarly, the IS-IS local RIB exposes the reachable prefixes in the IS-IS routing domain. Exposure of the LSDB and local RIB may also reveal information beyond the scope of the IS-IS router; this may be undesirable, since such exposure may facilitate other attacks. Additionally, the complete IP network topology -- and, if deployed, the TE topology of the IS-IS domain -- can be reconstructed from the LSDB. Though not as straightforward, the IS-IS local RIB can also be exploited to discover topological information. Network operators may consider their topologies to be sensitive confidential data.

For IS-IS authentication, configuration is supported via the specification of a key chain [RFC8177] or the direct specification of a key and authentication algorithm. Hence, authentication configuration using the "key-chain" case in the "authentication-type" container inherits the security considerations of [RFC8177]. This includes considerations with respect to the local storage and handling of authentication keys.

Some of the RPC operations in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations. These are the operations and their sensitivity/vulnerability:

The IS-IS YANG module supports the "clear-adjacency" and "clear-database" RPCs. If access to either of these is compromised, they can be exploited to mount DoS attacks due to the resultant network outages.

The actual authentication key data (whether locally specified or part of a key chain) is sensitive and needs to be kept secret from unauthorized parties; compromise of the key data would allow an attacker to forge IS-IS traffic that would be accepted as authentic, potentially compromising the entire IS-IS domain.

The model describes several notifications. Implementations must rate-limit the generation of these notifications to avoid creating significant notification load. Otherwise, this notification load may negatively affect system stability and may be exploited as an attack vector.

## 8. IANA Considerations

The IANA has assigned the following URI in the "IETF XML Registry" [RFC3688].

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

This document also adds the following YANG module name in the "YANG Module Names" registry [RFC6020]:

Name: ietf-isis  
Maintained by IANA? N  
Namespace: urn:ietf:params:xml:ns:yang:ietf-isis  
Prefix: isis  
Reference: RFC 9130

## 9. References

### 9.1. Normative References

[ISO-10589]

ISO, "Intermediate System to Intermediate System intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service (ISO 8473)", International Standard 10589: 2002, Second Edition, 2002, <<https://www.iso.org/standard/30932.html>>.

[RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, DOI 10.17487/RFC1195, December 1990, <<https://www.rfc-editor.org/info/rfc1195>>.

[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>>.

[RFC4090] Pan, P., Ed., Swallow, G., Ed., and A. Atlas, Ed., "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, DOI 10.17487/RFC4090, May 2005, <<https://www.rfc-editor.org/info/rfc4090>>.

[RFC5029] Vasseur, JP. and S. Previdi, "Definition of an IS-IS Link Attribute Sub-TLV", RFC 5029, DOI 10.17487/RFC5029, September 2007, <<https://www.rfc-editor.org/info/rfc5029>>.

[RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)", RFC 5120, DOI 10.17487/RFC5120, February 2008, <<https://www.rfc-editor.org/info/rfc5120>>.

[RFC5130] Previdi, S., Shand, M., Ed., and C. Martin, "A Policy Control Mechanism in IS-IS Using Administrative Tags", RFC 5130, DOI 10.17487/RFC5130, February 2008, <<https://www.rfc-editor.org/info/rfc5130>>.

[RFC5286] Atlas, A., Ed. and A. Zinin, Ed., "Basic Specification for IP Fast Reroute: Loop-Free Alternates", RFC 5286, DOI 10.17487/RFC5286, September 2008, <<https://www.rfc-editor.org/info/rfc5286>>.

- [RFC5301] McPherson, D. and N. Shen, "Dynamic Hostname Exchange Mechanism for IS-IS", RFC 5301, DOI 10.17487/RFC5301, October 2008, <<https://www.rfc-editor.org/info/rfc5301>>.
- [RFC5302] Li, T., Smit, H., and T. Przygienda, "Domain-Wide Prefix Distribution with Two-Level IS-IS", RFC 5302, DOI 10.17487/RFC5302, October 2008, <<https://www.rfc-editor.org/info/rfc5302>>.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <<https://www.rfc-editor.org/info/rfc5305>>.
- [RFC5307] Kompella, K., Ed. and Y. Rekhter, Ed., "IS-IS Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 5307, DOI 10.17487/RFC5307, October 2008, <<https://www.rfc-editor.org/info/rfc5307>>.
- [RFC5308] Hopps, C., "Routing IPv6 with IS-IS", RFC 5308, DOI 10.17487/RFC5308, October 2008, <<https://www.rfc-editor.org/info/rfc5308>>.
- [RFC5443] Jork, M., Atlas, A., and L. Fang, "LDP IGP Synchronization", RFC 5443, DOI 10.17487/RFC5443, March 2009, <<https://www.rfc-editor.org/info/rfc5443>>.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <<https://www.rfc-editor.org/info/rfc5880>>.
- [RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881, DOI 10.17487/RFC5881, June 2010, <<https://www.rfc-editor.org/info/rfc5881>>.
- [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>>.
- [RFC6119] Harrison, J., Berger, J., and M. Bartlett, "IPv6 Traffic Engineering in IS-IS", RFC 6119, DOI 10.17487/RFC6119, February 2011, <<https://www.rfc-editor.org/info/rfc6119>>.
- [RFC6232] Wei, F., Qin, Y., Li, Z., Li, T., and J. Dong, "Purge Originator Identification TLV for IS-IS", RFC 6232, DOI 10.17487/RFC6232, May 2011, <<https://www.rfc-editor.org/info/rfc6232>>.
- [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>>.
- [RFC7490] Bryant, S., Filsfils, C., Previdi, S., Shand, M., and N. So, "Remote Loop-Free Alternate (LFA) Fast Reroute (FRR)", RFC 7490, DOI 10.17487/RFC7490, April 2015,

<<https://www.rfc-editor.org/info/rfc7490>>.

- [RFC7794] Ginsberg, L., Ed., Decraene, B., Previdi, S., Xu, X., and U. Chunduri, "IS-IS Prefix Attributes for Extended IPv4 and IPv6 Reachability", RFC 7794, DOI 10.17487/RFC7794, March 2016, <<https://www.rfc-editor.org/info/rfc7794>>.
- [RFC7917] Sarkar, P., Ed., Gredler, H., Hegde, S., Litkowski, S., and B. Decraene, "Advertising Node Administrative Tags in IS-IS", RFC 7917, DOI 10.17487/RFC7917, July 2016, <<https://www.rfc-editor.org/info/rfc7917>>.
- [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>>.
- [RFC7981] Ginsberg, L., Previdi, S., and M. Chen, "IS-IS Extensions for Advertising Router Information", RFC 7981, DOI 10.17487/RFC7981, October 2016, <<https://www.rfc-editor.org/info/rfc7981>>.
- [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>>.
- [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>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.
- [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>>.
- [RFC8405] Decraene, B., Litkowski, S., Gredler, H., Lindem, A., Francois, P., and C. Bowers, "Shortest Path First (SPF) Back-Off Delay Algorithm for Link-State IGPs", RFC 8405, DOI 10.17487/RFC8405, June 2018, <<https://www.rfc-editor.org/info/rfc8405>>.
- [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>>.

- [RFC8570] Ginsberg, L., Ed., Previdi, S., Ed., Giacalone, S., Ward, D., Drake, J., and Q. Wu, "IS-IS Traffic Engineering (TE) Metric Extensions", RFC 8570, DOI 10.17487/RFC8570, March 2019, <<https://www.rfc-editor.org/info/rfc8570>>.
- [RFC8706] Ginsberg, L. and P. Wells, "Restart Signaling for IS-IS", RFC 8706, DOI 10.17487/RFC8706, February 2020, <<https://www.rfc-editor.org/info/rfc8706>>.
- [RFC9314] Jethanandani, M., Ed., Rahman, R., Ed., Zheng, L., Ed., Pallagatti, S., and G. Mirsky, "YANG Data Model for Bidirectional Forwarding Detection (BFD)", RFC 9314, DOI 10.17487/RFC9314, September 2022, <<https://www.rfc-editor.org/info/rfc9314>>.
- [W3C.REC-xml-20081126] Bray, T., Paoli, J., Sperberg-McQueen, M., Maler, E., and F. Yergeau, "Extensible Markup Language (XML) 1.0 (Fifth Edition)", World Wide Web Consortium Recommendation REC-xml-20081126, November 2008, <<https://www.w3.org/TR/2008/REC-xml-20081126>>.

## 9.2. Informative References

- [RFC7812] Atlas, A., Bowers, C., and G. Enyedi, "An Architecture for IP/LDP Fast Reroute Using Maximally Redundant Trees (MRT-FRR)", RFC 7812, DOI 10.17487/RFC7812, June 2016, <<https://www.rfc-editor.org/info/rfc7812>>.
- [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>>.
- [RFC8792] Watsen, K., Auerswald, E., Farrel, A., and Q. Wu, "Handling Long Lines in Content of Internet-Drafts and RFCs", RFC 8792, DOI 10.17487/RFC8792, June 2020, <<https://www.rfc-editor.org/info/rfc8792>>.
- [SR-TI-LFA] Litkowski, S., Bashandy, A., Filsfils, C., Francois, P., Decraene, B., and D. Voyer, "Topology Independent Fast Reroute using Segment Routing", Work in Progress, Internet-Draft, draft-ietf-rtgwg-segment-routing-ti-lfa-08, 21 January 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-rtgwg-segment-routing-ti-lfa-08>>.

## Appendix A. Example of IS-IS Configuration in XML

This appendix gives an example of the configuration of an IS-IS instance on a device. The example is written in XML [W3C.REC-xml-20081126].

```
<?xml version="1.0" encoding="utf-8"?>
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <routing xmlns="urn:ietf:params:xml:ns:yang:ietf-routing">
    <name>SLI</name>
    <router-id>192.0.2.1</router-id>
    <control-plane-protocols>
      <control-plane-protocol>
        <name>IS-IS-example</name>
        <description/>
        <type>
          <type xmlns:isis="urn:ietf:params:xml:ns:yang:ietf-isis">
```

```

isis:isis
</type>
</type>
<isis xmlns="urn:ietf:params:xml:ns:yang:ietf-isis">
  <enabled>true</enabled>
  <level-type>level-2</level-type>
  <system-id>87FC.FCDF.4432</system-id>
  <area-address>49.0001</area-address>
  <mpls>
    <te-rid>
      <ipv4-router-id>192.0.2.1</ipv4-router-id>
    </te-rid>
  </mpls>
  <lsp-lifetime>65535</lsp-lifetime>
  <lsp-refresh>65000</lsp-refresh>
  <metric-type>
    <value>wide-only</value>
  </metric-type>
  <default-metric>
    <value>111111</value>
  </default-metric>
  <address-families>
    <address-family-list>
      <address-family>ipv4</address-family>
      <enabled>true</enabled>
    </address-family-list>
    <address-family-list>
      <address-family>ipv6</address-family>
      <enabled>true</enabled>
    </address-family-list>
  </address-families>
  <interfaces>
    <interface>
      <name>Loopback0</name>
      <tag>200</tag>
      <metric>
        <value>0</value>
      </metric>
      <passive>true</passive>
    </interface>
    <interface>
      <name>Eth1</name>
      <level-type>level-2</level-type>
      <interface-type>point-to-point</interface-type>
      <metric>
        <value>167890</value>
      </metric>
    </interface>
  </interfaces>
</isis>
</control-plane-protocol>
</control-plane-protocols>
</routing>
<interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
  <interface>
    <name>Loopback0</name>
    <description/>
    <type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type">
      ianaift:softwareLoopback
    </type>
    <link-up-down-trap-enable>enabled</link-up-down-trap-enable>
    <ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
      <address>
        <ip>192.0.2.1</ip>
        <prefix-length>32</prefix-length>
      </address>
    </ipv4>
  </interface>
</interfaces>

```

```

</ipv4>
<ipv6 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
  <address>
    <ip>2001:db8::1</ip>
    <prefix-length>128</prefix-length>
  </address>
</ipv6>
</interface>
<interface>
  <name>Eth1</name>
  <description/>
  <type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type">
    ianaift:ethernetCsmacd
  </type>
  <link-up-down-trap-enable>enabled</link-up-down-trap-enable>
  <ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
    <address>
      <ip>198.51.100.1</ip>
      <prefix-length>30</prefix-length>
    </address>
  </ipv4>
  <ipv6 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
    <address>
      <ip>2001:db8:0:0:ff::1</ip>
      <prefix-length>64</prefix-length>
    </address>
  </ipv6>
</interface>
</interfaces>
</data>

```

#### Acknowledgments

The authors would like to thank Tom Petch, Alvaro Retana, Stewart Bryant, Barry Leiba, Benjamin Kaduk, Adam Roach, and Roman Danyliw for their review and comments.

#### Contributors

The authors would like to thank Kiran Agrahara Sreenivasa, Dean Bogdanovic, Yingzhen Qu, Yi Yang, and Jeff Tantsura for their major contributions to this document.

#### Authors' Addresses

Stephane Litkowski (editor)  
Cisco Systems  
Email: slitkows.ietf@gmail.com

Derek Yeung  
Arrcus, Inc.  
2077 Gateway Place, Suite 400  
San Jose, CA 95110  
United States of America  
Email: derek@arrcus.com

Acee Lindem  
Cisco Systems  
301 Midenhall Way  
Cary, NC 27513  
United States of America  
Email: acee@cisco.com

Jeffrey Zhang  
Juniper Networks  
10 Technology Park Drive  
Westford, MA 01886  
United States of America  
Email: zzhang@juniper.net

Ladislav Lhotka  
CZ.NIC  
Email: ladislav.lhotka@nic.cz