IS-IS Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 1, 2017

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

March 30, 2017

YANG Data Model for IS-IS protocol
draft-ietf-isis-yang-isis-cfg-17

Abstract

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

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Status of This Memo

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

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

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

This Internet-Draft will expire on October 1, 2017.

Copyright Notice

Copyright (c) 2017 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 (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 3
  1.1. Tree diagram 3
2. Design of the Data Model 3
  2.1. IS-IS Configuration 11
  2.2. Multitopology Parameters 11
  2.3. Per-Level Parameters 11
  2.4. Per-Interface Parameters 13
  2.5. Authentication Parameters 16
  2.6. IGP/LDP synchronization 16
  2.7. ISO parameters 16
  2.8. IP FRR 16
  2.9. Operational States 17
3. RPC Operations 17
4. Notifications 18
5. Interaction with Other YANG Modules 22
6. IS-IS YANG Module 23
7. Security Considerations 99
8. Contributors 100
9. Acknowledgements 100
10. IANA Considerations 100
11. Change log for ietf-isis YANG module 100
  11.1. From version -16 to version -17 100
  11.2. From version -15 to version -16 100
  11.3. From version -14 to version -15 101
  11.4. From version -13 to version -14 101
  11.5. From version -12 to version -13 101
  11.6. From version -09 to version -12 101
  11.7. From version -08 to version -09 102
  11.8. From version -07 to version -08 102
  11.9. From version -05 to version -07 102
  11.10. From version -03 to version -05 102

11.11. From version -02 to version -03 103
11.12. From version -01 to version -02 103
11.13. From version -00 to version -01 103
12. Normative References 104
Appendix A. Example of IS-IS configuration in XML 105
Authors' Addresses 107

1. Introduction

This document defines a YANG data model for IS-IS routing protocol.

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

1.1. Tree diagram

A simplified graphical representation of the data model is presented in Section 2.

The meaning of the symbols in these diagrams is as follows:

- o Brackets "[" and "]" enclose list keys.
- o Curly braces "{" and "}" contain names of optional features that make the corresponding node conditional.
- o Abbreviations before data node names: "rw" means configuration (read-write), and "ro" state data (read-only).
- o Symbols after data node names: "?" means an optional node and "*" denotes a "list" or "leaf-list".
- o Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").
- o Ellipsis ("...") stands for contents of subtrees that are not shown.

2. Design of the Data Model

The IS-IS YANG module is divided in two main "isis" containers that are augmenting the "control-plane-protocol" lists in ietf-routing module with specific IS-IS parameters.

One container contains the writable parameters, while the other contains the operational states.

The figure below describes the overall structure of the isis YANG module:

```
module: ietf-isis
augment /rt:routing-state/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
augment /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol
:
   +--rw isis
      +--rw enable?                   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 mpls
      |  +--rw ipv4-router-id?   inet:ipv4-address {ipv4-router-id}?
      |  +--rw ipv6-router-id?   inet:ipv6-address {ipv6-router-id}?
      |  +--rw ldp
      |        ...
      +--rw auto-cost {auto-cost}?
      |  +--rw reference-bandwidth?   uint32
      |  +--rw enable?                boolean
      +--rw lsp-mtu?                  uint16
      +--rw lsp-lifetime?             uint16
      +--rw lsp-refresh?              rt-types:timer-value-seconds16 {ls
p-refresh}?
      +--rw graceful-restart {graceful-restart}?
      |  +--rw enable?             boolean
      |  +--rw restart-interval?   rt-types:timer-value-seconds16
      |  +--rw helper-enable?      boolean
      +--rw nsr {nsr}?
      |  +--rw enable?   boolean
      +--rw node-tags {node-tag}?
      |  +--rw node-tag* [tag]
      |        ...
      +--rw authentication
      |  +--rw (authentication-type)?
      |  |     ...
      |  +--rw level-1
      |  |     ...
      |  +--rw level-2
      |        ...
      +--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 afs {nlpid-control}?
      |  +--rw af* [af]
      |        ...
      +--rw preference
      |  +--rw (granularity)?
      |        ...
      +--rw overload
      |  +--rw status?   boolean
      +--rw overload-max-metric {overload-max-metric}?
      |  +--rw timeout?   rt-types:timer-value-seconds16
      +--rw fast-reroute {fast-reroute}?
      |  +--rw lfa {lfa}?
      +--rw spf-control
      |  +--rw ietf-spf-delay {ietf-spf-delay}?
      |        ...
      +--rw topologies {multi-topology}?
      |  +--rw topology* [name]
      |        ...
      +--rw interfaces
         +--rw interface* [name]
               ...
augment /rt:routing-state/rt:control-plane-protocols/rt:control-plane-pr
otocol:
   +--ro isis
      +--ro enable?                   boolean {admin-control}?
      +--ro level-type?               level
      +--ro system-id?                system-id
      +--ro maximum-area-addresses?   uint8 {maximum-area-addresses}?
      +--ro area-address*             area-address
      +--ro mpls
      |  +--ro ipv4-router-id?   inet:ipv4-address {ipv4-router-id}?
      |  +--ro ipv6-router-id?   inet:ipv6-address {ipv6-router-id}?
      |  +--ro ldp
      |        ...
      +--ro auto-cost {auto-cost}?
      |  +--ro reference-bandwidth?   uint32
      |  +--ro enable?                boolean
      +--ro lsp-mtu?                  uint16
      +--ro lsp-lifetime?             uint16
      +--ro lsp-refresh?              rt-types:timer-value-seconds16 {ls
```

```
p-refresh}?
      +--ro graceful-restart {graceful-restart}?
```

```
      |  +--ro enable?             boolean
      |  +--ro restart-interval?   rt-types:timer-value-seconds16
      |  +--ro helper-enable?      boolean
      +--ro nsr {nsr}?
      |  +--ro enable?   boolean
      +--ro node-tags {node-tag}?
      |  +--ro node-tag* [tag]
      |        ...
      +--ro authentication
      |  +--ro (authentication-type)?
      |  |     ...
      |  +--ro level-1
      |  |     ...
      |  +--ro level-2
      |        ...
      +--ro metric-type
      |  +--ro value?     enumeration
      |  +--ro level-1
      |  |     ...
      |  +--ro level-2
      |        ...
      +--ro default-metric
      |  +--ro value?     wide-metric
      |  +--ro level-1
      |  |     ...
      |  +--ro level-2
      |        ...
      +--ro afs {nlpid-control}?
      |  +--ro af* [af]
      |        ...
      +--ro preference
      |  +--ro (granularity)?
      |        ...
      +--ro overload
      |  +--ro status?   boolean
      +--ro overload-max-metric {overload-max-metric}?
      |  +--ro timeout?   rt-types:timer-value-seconds16
      +--ro fast-reroute {fast-reroute}?
      |  +--ro lfa {lfa}?
      |  +--ro protected-routes
      |  |     ...
      |  +--ro nonprotected-routes
      |  |     ...
      |  +--ro protection-statistics* [frr-protection-method]
      |        ...
      +--ro spf-control
      |  +--ro ietf-spf-delay {ietf-spf-delay}?
      |        ...
```

```
      +--ro topologies* [name]
      |  +--ro name         leafref
      |  +--ro local-rib
      |        ...
      +--ro local-rib
      |  +--ro route* [prefix]
      |        ...
      +--ro system-counters
      |  +--ro level* [level]
      |        ...
      +--ro interfaces
      |  +--ro interface* [interface]
      |        ...
      +--ro spf-log
      |  +--ro event* [id]
      |        ...
      +--ro lsp-log
      |  +--ro event* [id]
      |        ...
      +--ro hostnames
      |  +--ro hostname* [system-id]
      |        ...
      +--ro database
         +--ro level-db* [level]
               ...
rpcs:
   +---x clear-adjacency
   |  +--ro input
   |     +--ro routing-protocol-instance-name    instance-state-ref
   |     +--ro level?                            level
   |     +--ro interface?                        string
   +---x clear-database
      +--ro input
         +--ro routing-protocol-instance-name    instance-state-ref
         +--ro level?                            level
notifications:
   +---n database-overload
   |  +--ro routing-instance?        string
   |  +--ro routing-protocol-name?   string
   |  +--ro isis-level?              level
   |  +--ro overload?                enumeration
   +---n lsp-too-large
   |  +--ro routing-instance?        string
   |  +--ro routing-protocol-name?   string
   |  +--ro isis-level?              level
   |  +--ro interface-name?          string
   |  +--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-instance?        string
   |  +--ro routing-protocol-name?   string
   |  +--ro isis-level?              level
   |  +--ro interface-name?          string
   |  +--ro interface-level?         level
   |  +--ro extended-circuit-id?     extended-circuit-id
   |  +--ro state?                   if-state-type
   +---n corrupted-lsp-detected
   |  +--ro routing-instance?        string
   |  +--ro routing-protocol-name?   string
   |  +--ro isis-level?              level
   |  +--ro lsp-id?                  lsp-id
   +---n attempt-to-exceed-max-sequence
   |  +--ro routing-instance?        string
   |  +--ro routing-protocol-name?   string
   |  +--ro isis-level?              level
   |  +--ro lsp-id?                  lsp-id
   +---n id-len-mismatch
   |  +--ro routing-instance?        string
   |  +--ro routing-protocol-name?   string
   |  +--ro isis-level?              level
   |  +--ro interface-name?          string
   |  +--ro interface-level?         level
   |  +--ro extended-circuit-id?     extended-circuit-id
   |  +--ro pdu-field-len?           uint8
   |  +--ro raw-pdu?                 binary
   +---n max-area-addresses-mismatch
   |  +--ro routing-instance?        string
   |  +--ro routing-protocol-name?   string
   |  +--ro isis-level?              level
   |  +--ro interface-name?          string
   |  +--ro interface-level?         level
   |  +--ro extended-circuit-id?     extended-circuit-id
   |  +--ro max-area-addresses?      uint8
   |  +--ro raw-pdu?                 binary
   +---n own-lsp-purge
   |  +--ro routing-instance?        string
   |  +--ro routing-protocol-name?   string
   |  +--ro isis-level?              level
   |  +--ro interface-name?          string
   |  +--ro interface-level?         level
   |  +--ro extended-circuit-id?     extended-circuit-id
   |  +--ro lsp-id?                  lsp-id
   +---n sequence-number-skipped
   |  +--ro routing-instance?        string
```

```
   |  +--ro routing-protocol-name?   string
   |  +--ro isis-level?              level
   |  +--ro interface-name?          string
   |  +--ro interface-level?         level
   |  +--ro extended-circuit-id?     extended-circuit-id
   |  +--ro lsp-id?                  lsp-id
   +---n authentication-type-failure
   |  +--ro routing-instance?        string
   |  +--ro routing-protocol-name?   string
   |  +--ro isis-level?              level
   |  +--ro interface-name?          string
   |  +--ro interface-level?         level
   |  +--ro extended-circuit-id?     extended-circuit-id
   |  +--ro raw-pdu?                 binary
   +---n authentication-failure
   |  +--ro routing-instance?        string
   |  +--ro routing-protocol-name?   string
   |  +--ro isis-level?              level
   |  +--ro interface-name?          string
   |  +--ro interface-level?         level
   |  +--ro extended-circuit-id?     extended-circuit-id
   |  +--ro raw-pdu?                 binary
   +---n version-skew
   |  +--ro routing-instance?        string
   |  +--ro routing-protocol-name?   string
   |  +--ro isis-level?              level
   |  +--ro interface-name?          string
   |  +--ro interface-level?         level
   |  +--ro extended-circuit-id?     extended-circuit-id
   |  +--ro protocol-version?        uint8
   |  +--ro raw-pdu?                 binary
   +---n area-mismatch
   |  +--ro routing-instance?        string
   |  +--ro routing-protocol-name?   string
   |  +--ro isis-level?              level
   |  +--ro interface-name?          string
   |  +--ro interface-level?         level
   |  +--ro extended-circuit-id?     extended-circuit-id
   |  +--ro raw-pdu?                 binary
   +---n rejected-adjacency
   |  +--ro routing-instance?        string
   |  +--ro routing-protocol-name?   string
   |  +--ro isis-level?              level
   |  +--ro interface-name?          string
   |  +--ro interface-level?         level
   |  +--ro extended-circuit-id?     extended-circuit-id
   |  +--ro raw-pdu?                 binary
   |  +--ro reason?                  string
```

```
   +---n protocols-supported-mismatch
   |  +--ro routing-instance?        string
   |  +--ro routing-protocol-name?   string
   |  +--ro isis-level?              level
   |  +--ro interface-name?          string
   |  +--ro interface-level?         level
   |  +--ro extended-circuit-id?     extended-circuit-id
   |  +--ro raw-pdu?                 binary
   |  +--ro protocols*               uint8
   +---n lsp-error-detected
   |  +--ro routing-instance?        string
   |  +--ro routing-protocol-name?   string
   |  +--ro isis-level?              level
   |  +--ro interface-name?          string
   |  +--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-instance?        string
   |  +--ro routing-protocol-name?   string
   |  +--ro isis-level?              level
   |  +--ro interface-name?          string
   |  +--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-instance?        string
   |  +--ro routing-protocol-name?   string
   |  +--ro isis-level?              level
   |  +--ro interface-name?          string
   |  +--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-instance?        string
      +--ro routing-protocol-name?   string
      +--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 container is divided in:

o Global parameters.

o Per interface configuration (see Section 2.4).

Additional modules may be created this to support any additional parameters. These additional modules should augment the ietf-isis module.

The model implements features, thus some of the configuration statement becomes 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 shutdown a particular IS-IS instance.

The global configuration contains usual IS-IS parameters such as lsp-mtu, lsp-lifetime, lsp-refresh, default-metric...

2.2. Multitopology Parameters

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

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

The "name" used in the topology list should refer to an existing RIB of the device.

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

Multiple address families (like IPv4 or IPv6) can also be activated within the default topology. This can be achieved using the "afs" container (requiring "nlpid-control" feature to be advertised).

2.3. Per-Level Parameters

Some parameters allow a per level configuration. In this case, the parameter is modeled as a container with three configuration locations:

- o a top level container: corresponds to level-1-2, so the configuration applies to both levels.

- o a level-1 container: corresponds to level-1 specific parameters.

- o a level-2 container: corresponds to level-2 specific parameters.

```
+--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>
    <level-2>
        <value>200</value>
    </level-2>
</priority>
```

An implementation SHOULD prefer a level specific parameter over a level-all parameter. As example, if the priority is 100 for the level-1, 200 for the level-2 and 250 for the top level configuration, the implementation should use 100 for the level-1 and 200 for the level-2.

Some parameters like "overload bit" and "route preference" are not modeled to support a per level configuration. If an implementation supports per level configuration for such parameter, this implementation SHOULD augment the current model by adding both level-1 and level-2 containers and SHOULD 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 augment IS-IS routing protocol when used";
    }
    description
     "This augments 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 the non support of the level-1 and level-2 containers.

Finally, if an implementation supports per level configuration but does not support the level-1-2 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 defined in the "ietf-interfaces" YANG model.

Each interface has some interface-specific parameters that may have a different per level value as described in previous section. An interface-specific parameter always override an IS-IS global parameter.

Some parameters like 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 level-type?                level
            +--rw lsp-pacing-interval?       rt-types:timer-value-millis
econds
            +--rw lsp-retransmit-interval?   rt-types:timer-value-second
s16
            +--rw passive?                   boolean
            +--rw csnp-interval?             rt-types:timer-value-second
s16
            +--rw hello-padding
            |  +--rw enable?   boolean
            +--rw mesh-group-enable?         mesh-group-state
            +--rw mesh-group?                uint8
            +--rw interface-type?            interface-type
            +--rw enable?                    boolean {admin-control}?
            +--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-re
f
            |  |     +--:(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-re
f
            |        +--:(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 enable?   boolean
             +--rw afs {nlpid-control}?
             |  +--rw af* [af]
             |     +--rw af    identityref
             +--rw mpls
             |  +--rw ldp
             |     +--rw igp-sync?   boolean {ldp-igp-sync}?
             +--rw fast-reroute {fast-reroute}?
             |  +--rw lfa {lfa}?
             |     +--rw candidate-disabled?   boolean
             |     +--rw enable?               boolean
             |     +--rw remote-lfa {remote-lfa}?
             |     |  +--rw enable?   boolean
             |     +--rw level-1
             |     |  +--rw candidate-disabled?   boolean
             |     |  +--rw enable?               boolean
             |     |  +--rw remote-lfa {remote-lfa}?
             |     |     +--rw enable?   boolean
             |     +--rw level-2
             |        +--rw candidate-disabled?   boolean
             |        +--rw enable?               boolean
             |        +--rw remote-lfa {remote-lfa}?
             |           +--rw enable?   boolean
             +--rw topologies {multi-topology}?
                +--rw topology* [name]
                   +--rw name      leafref
                   +--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 enables authentication configuration through the IETF key-chain module ([I-D.ietf-rtgwg-yang-key-chain]). The IS-IS module imports the "ietf-key-chain" module and reuses some groupings to allow global and per interface configuration of authentication. If a global authentication is configured, an implementation SHOULD authenticate PSNPs, CSNPs and LSPs with the authentication parameters supplied. The authentication of hello PDUs can be activated on a per interface basis.

2.6. IGP/LDP synchronization

[RFC5443] defines a mechanism where IGP needs to be synchronized with LDP. An "ldp-igp-sync" feature has been defined in the model to support this mechanism. The "mpls/ldp/igp-sync" leaf under "interface" allows activation of the mechanism on a per interface basis. The "mpls/ldp/igp-sync" container in the global configuration is empty on purpose and is not required for the activation. The goal of this empty container is to allow easy augmentation with additional parameters like timers for example.

2.7. ISO parameters

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

This module augments interface configuration model to support ISO configuration parameters.

The clns-mtu can be defined under the interface.

2.8. IP FRR

This YANG model supports LFA ([RFC5286]) and remote LFA ([RFC7490]) as IP FRR techniques. The "fast-reroute" container may be augmented by other models to support other IPFRR flavors (MRT, TILFA ...).

The current version of the model supports activation of LFA and remote LFA at interface only. The global "lfa" container is present but kept empty to allow augmentation with vendor specific properties like policies.

Remote LFA is considered as a child of LFA. Remote LFA cannot be enabled if LFA is not enabled.

The "candidate-disabled" allows to mark an interface to not be used as a backup.

2.9. Operational States

An "isis" container provides operational states for IS-IS. This container is divided in multiple components:

- o system-counters : provides statistical informations about the global system.

- o interface : provides configuration state informations for each interface.

- o adjacencies: provides state informations about current IS-IS adjacencies.

- o spf-log: provides informations about SPF events on the node. This SHOULD be implemented as a wrapping buffer.

- o lsp-log: provides informations about LSP events on the node (reception of an LSP or modification of local LSP). This SHOULD be implemented as a wrapping buffer and an implementation MAY decide to log refresh LSPs or not.

- o local-rib: provides the IS-IS internal routing table view.

- o database: provides details on the current LSDB.

- o hostnames: provides informations about system-id to hostname mappings.

- o fast-reroute: provides informations about IP FRR.

3. RPC Operations

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

- o clear-isis-database: reset the content of a particular IS-IS database and restart database synchronization with the neighbors.

- o clear-isis-adjacency: restart a particular set of IS-IS adjacencies.

```
rpcs:
   +---x clear-adjacency
   |  +--ro input
   |     +--ro routing-protocol-instance-name    instance-state-ref
   |     +--ro level?                            level
   |     +--ro interface?                        string
   +---x clear-database
      +--ro input
         +--ro routing-protocol-instance-name    instance-state-ref
         +--ro level?                            level
```

4. Notifications

The "ietf-isis" module introduces some notifications :

database-overload : raised when overload condition is changed.

lsp-too-large : raised when the system tries to propagate a too large PDU.

corrupted-lsp-detected : raised when the system find that an LSP that was stored in 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 we receive a PDU with a different value for the System ID length.

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

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 a LSP with a parse error.

      adjacency-change : This notification is sent when an IS-IS
      adjacency moves to Up state or to Down state.

      lsp-received : This notification is sent when a LSP is received.

      lsp-generation : This notification is sent when a LSP is
      regenerated.

   notifications:
      +---n database-overload
      |  +--ro routing-instance?        string
      |  +--ro routing-protocol-name?   string
      |  +--ro isis-level?              level
      |  +--ro overload?                enumeration
      +---n lsp-too-large
      |  +--ro routing-instance?        string
      |  +--ro routing-protocol-name?   string
      |  +--ro isis-level?              level
      |  +--ro interface-name?          string
      |  +--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-instance?        string
      |  +--ro routing-protocol-name?   string
      |  +--ro isis-level?              level
      |  +--ro interface-name?          string
      |  +--ro interface-level?         level
      |  +--ro extended-circuit-id?     extended-circuit-id
      |  +--ro state?                   if-state-type
```

```
      +---n corrupted-lsp-detected
      |  +--ro routing-instance?        string
      |  +--ro routing-protocol-name?   string
      |  +--ro isis-level?              level
      |  +--ro lsp-id?                  lsp-id
      +---n attempt-to-exceed-max-sequence
      |  +--ro routing-instance?        string
      |  +--ro routing-protocol-name?   string
      |  +--ro isis-level?              level
      |  +--ro lsp-id?                  lsp-id
      +---n id-len-mismatch
      |  +--ro routing-instance?        string
      |  +--ro routing-protocol-name?   string
      |  +--ro isis-level?              level
      |  +--ro interface-name?          string
      |  +--ro interface-level?         level
      |  +--ro extended-circuit-id?     extended-circuit-id
      |  +--ro pdu-field-len?           uint8
      |  +--ro raw-pdu?                 binary
      +---n max-area-addresses-mismatch
      |  +--ro routing-instance?        string
      |  +--ro routing-protocol-name?   string
      |  +--ro isis-level?              level
      |  +--ro interface-name?          string
      |  +--ro interface-level?         level
      |  +--ro extended-circuit-id?     extended-circuit-id
      |  +--ro max-area-addresses?      uint8
      |  +--ro raw-pdu?                 binary
      +---n own-lsp-purge
      |  +--ro routing-instance?        string
      |  +--ro routing-protocol-name?   string
      |  +--ro isis-level?              level
      |  +--ro interface-name?          string
      |  +--ro interface-level?         level
      |  +--ro extended-circuit-id?     extended-circuit-id
      |  +--ro lsp-id?                  lsp-id
      +---n sequence-number-skipped
      |  +--ro routing-instance?        string
      |  +--ro routing-protocol-name?   string
      |  +--ro isis-level?              level
      |  +--ro interface-name?          string
      |  +--ro interface-level?         level
      |  +--ro extended-circuit-id?     extended-circuit-id
      |  +--ro lsp-id?                  lsp-id
      +---n authentication-type-failure
      |  +--ro routing-instance?        string
      |  +--ro routing-protocol-name?   string
      |  +--ro isis-level?              level
```

```
      |  +--ro interface-name?          string
      |  +--ro interface-level?         level
      |  +--ro extended-circuit-id?     extended-circuit-id
      |  +--ro raw-pdu?                 binary
      +---n authentication-failure
      |  +--ro routing-instance?        string
      |  +--ro routing-protocol-name?   string
      |  +--ro isis-level?              level
      |  +--ro interface-name?          string
      |  +--ro interface-level?         level
      |  +--ro extended-circuit-id?     extended-circuit-id
      |  +--ro raw-pdu?                 binary
      +---n version-skew
      |  +--ro routing-instance?        string
      |  +--ro routing-protocol-name?   string
      |  +--ro isis-level?              level
      |  +--ro interface-name?          string
      |  +--ro interface-level?         level
      |  +--ro extended-circuit-id?     extended-circuit-id
      |  +--ro protocol-version?        uint8
      |  +--ro raw-pdu?                 binary
      +---n area-mismatch
      |  +--ro routing-instance?        string
      |  +--ro routing-protocol-name?   string
      |  +--ro isis-level?              level
      |  +--ro interface-name?          string
      |  +--ro interface-level?         level
      |  +--ro extended-circuit-id?     extended-circuit-id
      |  +--ro raw-pdu?                 binary
      +---n rejected-adjacency
      |  +--ro routing-instance?        string
      |  +--ro routing-protocol-name?   string
      |  +--ro isis-level?              level
      |  +--ro interface-name?          string
      |  +--ro interface-level?         level
      |  +--ro extended-circuit-id?     extended-circuit-id
      |  +--ro raw-pdu?                 binary
      |  +--ro reason?                  string
      +---n protocols-supported-mismatch
      |  +--ro routing-instance?        string
      |  +--ro routing-protocol-name?   string
      |  +--ro isis-level?              level
      |  +--ro interface-name?          string
      |  +--ro interface-level?         level
      |  +--ro extended-circuit-id?     extended-circuit-id
      |  +--ro raw-pdu?                 binary
      |  +--ro protocols*               uint8
      +---n lsp-error-detected
```

```
      |  +--ro routing-instance?        string
      |  +--ro routing-protocol-name?   string
      |  +--ro isis-level?              level
      |  +--ro interface-name?          string
      |  +--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-instance?        string
      |  +--ro routing-protocol-name?   string
      |  +--ro isis-level?              level
      |  +--ro interface-name?          string
      |  +--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-instance?        string
      |  +--ro routing-protocol-name?   string
      |  +--ro isis-level?              level
      |  +--ro interface-name?          string
      |  +--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-instance?        string
         +--ro routing-protocol-name?   string
         +--ro isis-level?              level
         +--ro lsp-id?                  lsp-id
         +--ro sequence?                uint32
         +--ro send-timestamp?          yang:timestamp
```

5. Interaction with Other YANG Modules

The "isis" configuration container augments the "/rt:routing/rt:control-plane-protocols/control-plane-protocol" container of the ietf-routing [I-D.ietf-netmod-routing-cfg] module by defining IS-IS specific parameters.

The "isis" module augments "/if:interfaces/if:interface" with ISO specific parameters.

The "isis" operational state container augments the "/rt:routing-state/rt:control-plane-protocols/control-plane-protocol" container of the ietf-routing module by defining IS-IS specific operational states.

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

The modules defined in this document use some groupings from ietf-keychain [I-D.ietf-rtgwg-yang-key-chain].

6. IS-IS YANG Module

```
<CODE BEGINS> file "ietf-isis@2017-03-30.yang"

module ietf-isis {
  namespace "urn:ietf:params:xml:ns:yang:ietf-isis";

  prefix isis;

  import ietf-routing {
    prefix "rt";
  }

  import ietf-inet-types {
    prefix inet;
  }

  import ietf-yang-types {
    prefix yang;
  }

  import ietf-interfaces {
    prefix "if";
  }

  import ietf-key-chain {
    prefix "key-chain";
  }

  import ietf-routing-types {
    prefix "rt-types";
  }
```

```
    organization
     "IETF ISIS Working Group";

    contact
      "WG List:  &lt;mailto:isis-wg@ietf.org&gt;

      Editor:    Stephane Litkowski
            &lt;mailto:stephane.litkowski@orange.com&gt;

          Derek Yeung
            &lt;mailto:derek@arrcus.com&gt;
          Acee Lindem
            &lt;mailto:acee@cisco.com&gt;
          Jeffrey Zhang
            &lt;mailto:zzhang@juniper.net&gt;
          Ladislav Lhotka
            &lt;mailto:llhotka@nic.cz&gt;
          Yi Yang
            &lt;mailto:yiya@cisco.com&gt;
          Dean Bogdanovic
            &lt;mailto:deanb@juniper.net&gt;
          Kiran Agrahara Sreenivasa
            &lt;mailto:kkoushik@brocade.com&gt;
          Yingzhen Qu
            &lt;mailto:yiqu@cisco.com&gt;
          Jeff Tantsura
            &lt;mailto:jefftant.ietf@gmail.com&gt;

      ";

    description
      "The YANG module defines a generic configuration model for
      ISIS common across all of the vendor implementations.";

    revision 2017-03-30 {
      description
       "Initial revision.";
       reference "RFC XXXX";
    }

   /* Identities */

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

    identity isis-adjacency-change {
```

```
      description "Identity for the ISIS routing protocol
        adjacency state.";
    }

    identity clear-isis-database {
      description "Identity for the ISIS routing protocol
        database reset action.";
    }

    identity clear-isis-adjacency {
      description "Identity for the ISIS routing protocol
        adjacency reset action.";
    }

    identity lsp-log-reason {
      description "Base identity for an LSP change
               log reason.";
    }

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

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


    /* Feature definitions */

    feature ietf-spf-delay {
      description
       "Support of IETF SPF delay algorithm.";
    }
    feature bfd {
      description
       "Support of BFD for IS-IS links.";
    }
    feature key-chain {
      description
        "Support of keychain for authentication.";
    }
```

```
    feature segment-routing {
      description
       "Support of segment-routing.";
    }
    feature node-flag {
      description
       "Support of node-flag advertisement
       as prefix attribute";
    }
    feature node-tag {
      description
        "Support of node tag.";
    }
    feature ldp-igp-sync {
      description
        "Support of RFC5443.";
    }
    feature fast-reroute {
      description
        "Support of IPFRR.";
    }
    feature nsr {
      description
       "Support of
          Non Stop Routing.";
    }
    feature  lfa {
      description
        "Support of Loop Free Alternates.";
    }
    feature remote-lfa {
      description
        "Support of remote Loop Free Alternates.";
    }

    feature overload-max-metric {
      description
        "Support of overload by setting
         all links to max metric.";
    }
    feature prefix-tag {
      description
        "Add 32bit tag to prefixes";
    }
    feature prefix-tag64 {
      description
        "Add 64bit tag to prefixes";
    }
```

```
    feature auto-cost {
      description
        "Use an automated assignment of metrics.";
    }
    feature ipv4-router-id {
      description
       "Support of IPv4 router ID configuration under ISIS.";
    }

    feature ipv6-router-id {
      description
       "Support of IPv6 router ID configuration under ISIS.";
    }

    feature multi-topology {
      description
       "Multitopology routing support.";
    }
    feature nlpid-control {
      description
       "This feature controls the advertisement
        of support NLPID within ISIS configuration.";
    }
    feature graceful-restart {
      description
        "Graceful restart support as per RFC5306.";
    }

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

    feature maximum-area-addresses {
      description
        "Support of maximum-area-addresses config.";
    }

    feature admin-control {
      description
       "Control administrative state of ISIS.";
    }

    /* Type definitions */


    typedef instance-state-ref {
      type leafref {
```

```
        path "/rt:routing-state/"
        +"rt:control-plane-protocols/rt:control-plane-protocol/"
        +"rt:name";
      }
      description
         "This type is used for leaves that reference state data of
          an ISIS protocol instance.";
    }

    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 on the interface.
       This is affecting the type of hello
       message that would be used.";

    }

    typedef level {
      type enumeration {
        enum "level-1" {
          description
           "This enum describes L1 only capability.";
        }
```

```
        enum "level-2" {
          description
           "This enum describes L2 only capability.";
        }
        enum "level-all" {
          description
           "This enum describes both levels capability.";
        }
      }
      default "level-all";
      description
       "This type defines ISIS level of an object.";

    }

    typedef adj-state-type {
       type enumeration {
          enum "Up" {
            description
             "This state describes that
              adjacency is established.";
          }
          enum "Down" {
            description
             "This state describes that
              adjacency is NOT established.";
          }
          enum "Init" {
            description
             "This state describes that
              adjacency is establishing.";
          }
          enum "Failed" {
            description
             "This state describes that
              adjacency is failed.";
          }
        }
        description
        "This type defines states of an adjacency";
    }

    typedef if-state-type {
       type enumeration {
          enum "Up" {
            description
             "Up state.";
```

```
          }
          enum "Down" {
            description
             "Down state";
          }
        }
        description
        "This type defines states of an interface";
    }

    typedef level-number {
      type uint8 {
        range "1 .. 2";
      }
      description
       "This type defines a current ISIS 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 ISIS LSP ID using pattern,
        system id looks like : 0143.0438.AeF0.02-01";
    }

    typedef area-address {
      type string {
        pattern '[0-9A-Fa-f]{2}\.([0-9A-Fa-f]{4}\.){0,3}';
      }
      description
       "This type defines the area address format.";
    }

    typedef snpa {
      type string {
        length "0 .. 20";
      }
      description
       "This type defines Subnetwork Point
        of Attachement format.";
    }

    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 ISIS system id using pattern,
        system id looks like : 0143.0438.AeF0";
    }

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

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

    typedef mesh-group-state {
      type enumeration {
        enum "meshInactive" {
          description
           "Interface is not part of a mesh group.";
        }
        enum "meshSet" {
          description
           "Interface is part of a mesh group.";
        }
        enum "meshBlocked" {
          description
           "LSPs must not be flooded over that interface.";
        }
      }
      description
       "This type describes meshgroup state of an interface";
    }


    /* Grouping definitions for configuration and ops state */
```

```
    grouping adjacency-state {
     container adjacencies {
        list adjacency {
          leaf neighbor-systype {
            type level;
            description
             "Type of 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
             "Circuit ID of the neighbor";
          }
          leaf neighbor-snpa {
            type snpa;
            description
             "SNPA of the neighbor";
          }
          leaf usage {
            type level;
            description
            "How is the adjacency used ?
            On a p2p link this might be level 1 and 2,
            but on a LAN, the usage will be level 1
            between peers at L1 or level 2 between
            peers at L2.";
          }
          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 DIS.";
```

```
          }
          leaf lastuptime {
            type yang:timestamp;
            description
             "When the adjacency most recently entered
              state 'up', 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 state 'up'.";
          }
          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 fast-reroute-global-state {
        container protected-routes {
          list af-stats {
            key "af prefix alternate";

            leaf af {
              type identityref {
                base rt-types:address-family;
              }
              description
               "Address-family";
            }
            leaf prefix {
              type string;
              description
               "Protected prefix.";
            }
            leaf alternate {
              type string;
```

```
              description
               "Alternate nexthop for the prefix.";
            }
            leaf alternate-type {
              type enumeration {
                enum equalcost {
                  description
                   "ECMP alternate.";
                }
                enum lfa {
                  description
                   "LFA alternate.";
                }
                enum remote-lfa {
                  description
                   "Remote LFA alternate.";
                }
                enum tunnel {
                  description
                   "Tunnel based alternate
                   (like RSVP-TE or GRE).";
                }
                enum ti-lfa {
                  description
                   "TI LFA alternate.";
                }
                enum mrt {
                  description
                   "MRT alternate.";
                }
                enum other {
                  description
                   "Unknown alternate type.";
                }
              }
              description
               "Type of alternate.";
            }
            leaf best {
              type boolean;
              description
               "describes if the alternate is the best one.";
            }
            leaf non-best-reason {
              type string;
              description
               "Information field to describe why the alternate
               is not best.";
```

```
            }
            leaf protection-available {
              type bits {
                bit nodeprotect {
                  position 0;
                  description
                  "Node protection available.";
                }
                bit linkprotect {
                  position 1;
                  description
                  "Link protection available.";
                }
                bit srlgprotect {
                  position 2;
                  description
                  "SRLG protection available.";
                }
                bit downstreamprotect {
                  position 3;
                  description
                  "Downstream protection available.";
                }
                bit other {
                  position 4;
                  description
                  "Other protection available.";
                }
              }
              description
              "Describes protection provided by the alternate.";
            }
            leaf alternate-metric1 {
              type uint32;
              description
               "Metric from PLR to destination
                through the alternate path.";
            }
            leaf alternate-metric2 {
              type uint32;
              description
               "Metric from PLR to the alternate node";
            }
            leaf alternate-metric3 {
              type uint32;
              description
               "Metric from alternate node to the destination";
            }
```

```
            description
             "Per AF statistics.";
          }
          description
           "List of prefixes that are protected.";
        }

        container nonprotected-routes {
           list af-stats {
            key "af prefix";

            leaf af {
              type identityref {
                base rt-types:address-family;
              }
              description
               "Address-family";
            }
            leaf prefix {
              type string;
              description
               "Protected prefix.";
            }
            description
             "Per AF statistics.";
          }
          description
           "List of prefixes that are not protected.";
        }

        list protection-statistics {
          key frr-protection-method;

          leaf frr-protection-method {
            type string;
            description
             "Protection method used.";
          }
          list af-stats {
            key af;

            leaf af {
              type identityref {
                base rt-types:address-family;
              }
              description
               "Address-family";
            }
```

```
            leaf total-routes {
              type uint32;
              description
               "Total prefixes.";
            }
            leaf unprotected-routes {
              type uint32;
              description
                "Total of prefixes who are
                not protected.";
            }
            leaf protected-routes {
              type uint32;
              description
                "Total of prefixes who are
                protected.";
            }
            leaf linkprotected-routes {
              type uint32;
              description
                "Total of prefixes who are
                link protected.";
            }
            leaf nodeprotected-routes {
              type uint32;
              description
                "Total of prefixes who are
                node protected.";
            }
            description
             "Per AF statistics.";
          }

          description
           "Global protection statistics.";
        }
        description
         "IPFRR states.";
    }

    grouping notification-instance-hdr {
      description
       "This group describes common instance specific
        data for notifications.";
      leaf routing-instance {
        type string;
        description
         "Describes the name of the routing-instance instance.";
```

```
      }
      leaf routing-protocol-name {
        type string;
        description
         "Describes the name of the ISIS instance.";
      }
      leaf isis-level {
        type level;
        description
         "Describes the ISIS level of the instance.";
      }
    }

    grouping notification-interface-hdr {
      description
       "This group describes common interface specific
        data for notifications.";
      leaf interface-name {
        type string;
        description
         "Describes the name of the ISIS interface.";
      }
      leaf interface-level {
        type level;
        description
         "Describes the ISIS level of the interface.";
      }
      leaf extended-circuit-id {
        type extended-circuit-id;
        description
         "Describes the extended circuit-id of the interface.";
      }
    }

    grouping route-content {
      description
       "This group add isis-specific route properties.";
      leaf metric {
        type uint32;
        description
         "This leaf describes ISIS metric of a route.";
      }
      leaf-list tag {
        type uint64;
        description
         "This leaf describes list of tags associated
          with the route. The leaf describes both
          32bits and 64bits tags.";
```

```
      }
      leaf route-type {
        type enumeration {
          enum l2-up-internal {
            description "Level 2 internal route
             and not leaked to a lower level";
          }
          enum l1-up-internal {
            description "Level 1 internal route
             and not leaked to a lower level";
          }
          enum l2-up-external {
            description "Level 2 external route
             and not leaked to a lower level";
          }
          enum l1-up-external {
            description "Level 1 external route
             and not leaked to a lower level";
          }
          enum l2-down-internal {
            description "Level 2 internal route
             and leaked to a lower level";
          }
          enum l1-down-internal {
            description "Level 1 internal route
             and leaked to a lower level";
          }
          enum l2-down-external {
            description "Level 2 external route
             and leaked to a lower level";
          }
          enum l1-down-external {
            description "Level 1 external route
             and leaked to a lower level";
          }
        }
        description
         "This leaf describes the type of ISIS route.";
      }
    }

    grouping admin-control {
      leaf enable {
          if-feature admin-control;
          type boolean;
          default true;
          description
               "Control the administrative
```

```
                state.";
      }
      description
       "Grouping for admin control.";
    }

    grouping fast-reroute-global-cfg {
      description
       "This group defines global
       configuration of IPFRR.";
      container lfa {
        if-feature lfa;
        description
          "This container may be
          augmented with global parameters
          for LFA.
          Creating the container has no effect on
          LFA activation.";
      }
    }

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

    grouping ietf-spf-delay-cfg {
      leaf initial-delay {
          type rt-types:timer-value-milliseconds;
          units msec;
          description
```

```
           "Delay used while in QUIET state.";
      }
      leaf short-delay {
          type rt-types:timer-value-milliseconds;
          units msec;
          description
           "Delay used while in SHORT_WAIT state.";
      }
      leaf long-delay {
          type rt-types:timer-value-milliseconds;
          units msec;
          description
           "Delay used while in LONG_WAIT state.";
      }

      leaf hold-down {
          type rt-types:timer-value-milliseconds;
          units msec;
          description
           "Timer used to consider an IGP stability period.";
      }
      leaf time-to-learn {
          type rt-types:timer-value-milliseconds;
          units msec;
          description
           "Duration used to learn all the IGP events
           related to a single component failure.";
      }

      description
       "Grouping for IETF SPF delay configuration.";
    }

    grouping ietf-spf-delay-state {
      leaf current-state {
          type enumeration {
              enum "QUIET" {
                  description "QUIET state";
              }
              enum "SHORT_WAIT" {
                  description "SHORT_WAIT state";
              }
              enum "LONG_WAIT" {
                  description "LONG_WAIT state";
              }
          }
          description
           "Current state of the algorithm.";
```

```
      }
      leaf remaining-time-to-learn {
          type rt-types:timer-value-milliseconds;
          units "msec";
          description
           "Remaining time until time-to-learn timer fires.";
      }
      leaf remaining-hold-down {
          type rt-types:timer-value-milliseconds;
          units "msec";
          description
           "Remaining time until hold-down timer fires.";
      }
      leaf last-event-received {
          type yang:timestamp;
          description
           "Time of last IGP event received";
      }
      leaf next-spf-time {
          type yang:timestamp;
          description
           "Time when next SPF has been scheduled.";
      }
      leaf last-spf-time {
          type yang:timestamp;
          description
           "Time of last SPF computation.";
      }
      description
       "Grouping for IETF SPF delay operational states.";
    }

     grouping local-rib {
      description "Local-rib grouping.";
      container local-rib {
        description "Local-rib.";
        list route {
          key "prefix";
          description "Routes";
          leaf prefix {
            type inet:ip-prefix;
            description "Destination prefix.";
          }
          container next-hops {
            description "All next hops for the route.";
            list next-hop {
              key "next-hop";
              description "List of next hop for the route";
```

```
              leaf outgoing-interface {
                type if:interface-ref;
                description
                  "Name of the outgoing interface.";
              }
              leaf next-hop {
               type inet:ip-address;
               description "Nexthop 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 isis-node-tag-cfg {
      description
       "ISIS node tag config.";
      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 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 describes the
                 authentication key.";
            }
            leaf crypto-algorithm {
              type identityref {
                  base key-chain:crypto-algorithm;
              }
              description
                "Cryptographic algorithm associated with key.";
            }
          }
          description
             "Choice of authentication.";
        }
       description
        "Grouping for global auth config.";
    }

    grouping metric-type-global-cfg {
      leaf value {
          type enumeration {
            enum wide-only {
              description
               "Advertise new metric style only
               (RFC5305)";
            }
            enum old-only {
              description
               "Advertise old metric style only
               (RFC1195)";
            }
            enum both {
              description "Advertise both metric
              styles";
            }
```

```
          }
          description
           "This leaf describes the type of metric
           to be generated.
           Wide-only means only new metric style
           is generated,
           old-only means that only old style metric
           is generated,
           and both means that both are advertised.
           This leaf is only affecting IPv4 metrics.";
        }
         description
        "Grouping for global metric style config.";
    }

    grouping default-metric-global-cfg {
      leaf value {
          type wide-metric;
          default "10";
          description
            "Value of the metric";
      }
      description
            "Grouping for global default metric config.";
    }


    grouping overload-global-cfg {
      leaf status {
          type boolean;
          description
           "This leaf defines the overload status.";
      }
            description
        "Grouping for overload bit config.";
    }

    grouping overload-max-metric-global-cfg {
      leaf timeout {
          type rt-types:timer-value-seconds16;
          units "seconds";
          description
           "This leaf defines the timeout in seconds
            of the overload condition.";
        }
        description
        "Grouping for overload-max-metric config.";
    }
```

```
    grouping route-preference-global-cfg {
      choice granularity {
              case detail {
                leaf internal {
                  type uint8;
                  description
                    "This leaf defines the protocol
                     preference for internal routes.";
                }
                leaf external {
                  type uint8;
                  description
                    "This leaf defines the protocol
                     preference for external routes.";
                }
              }
              case coarse {
                leaf default {
                  type uint8;
                  description
                    "This leaf defines the protocol
                     preference for all ISIS routes.";
                }
              }
              description
               "Choice for implementation of route preference.";
            }
      description
       "This grouping defines how route preference is configured.";
    }

    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
               "This leaf describes the
                 authentication key.";
            }
```

```
            leaf crypto-algorithm {
              type identityref {
                  base key-chain:crypto-algorithm;
              }
              description
                "Cryptographic algorithm associated with key.";
            }
          }
          description
             "Choice of authentication.";
        }
        description
         "Grouping for hello authentication.";
    }

    grouping hello-interval-cfg {
      leaf value {
      type rt-types:timer-value-seconds16;
      units "seconds";
      default 10;
      description
       "This leaf defines the interval of
       hello messages.";
      }

      description
       "Interval between
       hello messages.";
    }

    grouping hello-multiplier-cfg {
      leaf value {
      type uint16;
      description
       "This leaf defines the number of
       hello failed to be received before
       declaring the adjacency down.";
      }
      description
       "This grouping defines the number of
        hello failed to be received before
        declaring the adjacency down.";
    }

    grouping priority-cfg {
      leaf value {
          type uint8 {
            range "0 .. 127";
```

```
      }
      default 64;
      description
       "This leaf describes the priority of
       the interface
       for DIS election.";
      }

      description
       "This grouping leaf describes the
       priority of
       the interface
       for DIS election.";
    }

    grouping metric-cfg {
        leaf value {
          type wide-metric;
          description
           "Metric value.";
        }
        description
         "Grouping for interface metric";
    }

    grouping lfa-if-cfg {
      leaf candidate-disabled {
        type boolean;
        default false;
        description
         "Prevent the interface to be used as backup.";
      }
      leaf enable {
        type boolean;
        description
         "Activates LFA.
         This model assumes activation
         of per-prefix LFA.";
      }

      container remote-lfa {
        if-feature remote-lfa;
        leaf enable {
          type boolean;
          description
          "Activates rLFA.";
        }
        description
```

```
         "remote LFA configuration.";
      }
      description
       "Grouping for LFA
       interface configuration";
    }

    grouping isis-global-cfg {
      description
       "Defines the ISIS global configuration.";

          uses admin-control;

          leaf level-type {
            type level;
            default "level-all";
            description
             "This leaf describes the type of ISIS node.
             A node can be level-1-only, level-2-only
             or level-1-2.
             ";
          }

          leaf system-id {
            type system-id;
            description
             "This leaf defines the system-id of the node.";
          }

          leaf maximum-area-addresses {
            if-feature maximum-area-addresses;
            type uint8;
            default 3;
            description
             "Defines the maximum areas supported.";
          }

          leaf-list area-address {
            type area-address;
            description
             "List of areas supported by the
             protocol instance.";
          }

          container mpls {
            leaf ipv4-router-id {
              if-feature ipv4-router-id;
              type inet:ipv4-address;
```

```
              description
               "Router ID value that would be used in
               TLV 134.";
            }
            leaf ipv6-router-id {
              if-feature 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 igp-ldp-sync.";
              }
              description
               "LDP related configuration.";
            }
            description
              "This container handles mpls config.";
          }
          container auto-cost {
              if-feature auto-cost;
              leaf reference-bandwidth {
                type uint32;
                units "bps";
                description
                 "This leaf defines the bandwidth for calculating
                  metric.";
              }
              leaf enable {
                  type boolean;
                  default false;
                  description
                  "Enable/disable auto-cost.";
              }
              description
               "This container defines the auto-cost configuration.";
          }
          leaf lsp-mtu {
            type uint16;
            units "bytes";
            default 1492;
            description
             "This leaf describes the maximum size of a
```

```
              LSP PDU in bytes.";
          }
          leaf lsp-lifetime {
            type uint16 {
              range "1..65535";
              }
            units "seconds";
            description
             "This leaf describes the lifetime of the router
              LSP in seconds.";
          }
          leaf lsp-refresh {
            if-feature lsp-refresh;
            type rt-types:timer-value-seconds16;
            units "seconds";
            description
             "This leaf describes the refresh interval of the
              router LSP in seconds.";
          }
          container graceful-restart {
            if-feature graceful-restart;
            leaf enable {
              type boolean;
              description
               "Control enabling the feature.";
            }
            leaf restart-interval {
              type rt-types:timer-value-seconds16;
              units "seconds";
              description
              "Interval in seconds to attempt graceful restart prior
             to failing";
            }
            leaf helper-enable {
              type boolean;
              description
              "If enabled, the local router can act as restart helper.";
            }
            description
             "This container activates graceful restart.";
          }

          container nsr {
              if-feature nsr;
              description
              "Non-Stop Routing (NSR) config state.";
              leaf enable {
                  type boolean;
```

```
                description
                 "Enable/Disable NSR.";
            }
        }

        uses isis-node-tag-cfg;

        container authentication {
            uses authentication-global-cfg;

            container level-1 {
                uses authentication-global-cfg;
                description "level-1 specific cfg";
            }
            container level-2 {
                uses authentication-global-cfg;
                description "level-2 specific cfg";
            }
            description "authentication global cfg.
            It covers both LSPs and SNPs.";
        }

        container metric-type {
            uses metric-type-global-cfg;
            container level-1 {
                uses metric-type-global-cfg;
                description "level-1 specific cfg";
            }
            container level-2 {
                uses metric-type-global-cfg;
                description "level-2 specific cfg";
            }
            description "Metric style global cfg.";
        }

        container default-metric {
            uses default-metric-global-cfg;
            container level-1 {
                uses default-metric-global-cfg;
                description "level-1 specific cfg";
            }
            container level-2 {
                uses default-metric-global-cfg;
                description "level-2 specific cfg";
            }
            description "Default metric global cfg.";
        }
```

```
          container afs {
              if-feature nlpid-control;
              list af {
                key af;
                leaf af {
                  type identityref {
                    base rt-types:address-family;
                  }
                  description
                   "Address-family";
                }
                leaf enable {
                  type boolean;
                  description
                    "Describes the activation state of the
                     AF.";
                }
                description
                 "This list permits activation
                 of new address families.";
              }
              description
               "Container for address-families";
          }

          container preference {
              uses route-preference-global-cfg;
              description
               "This container defines the protocol preference.";
          }

          container overload {
              uses overload-global-cfg;
              description
             "This container describes if the router is
              set to overload state.";
          }

          container overload-max-metric {
              if-feature overload-max-metric;
              uses overload-max-metric-global-cfg;
              description
             "This container describes if the router is
              set to overload state using max-metric
              advertisement.";
          }
      }
```

```
    grouping isis-global-topologies-cfg {
     description
      "Per topology config.";
      container default-metric {
          uses default-metric-global-cfg;
          container level-1 {
              uses default-metric-global-cfg;
              description "level-1 specific cfg";
          }
          container level-2 {
              uses default-metric-global-cfg;
              description "level-2 specific cfg";
          }
          description "Default metric per
          topology cfg.";
      }
      uses isis-node-tag-cfg;
    }

    grouping isis-if-cfg {
      description
       "Grouping for interface cfg.";

      leaf level-type {
        type level;
        default "level-all";
        description
         "This leaf defines the associated ISIS
         level of the interface.";
      }
      leaf lsp-pacing-interval {
        type rt-types:timer-value-milliseconds;
        units "milliseconds";
        default 33;
        description
         "This leaf defines the interval between
          LSP transmissions in milli-seconds";
      }
      leaf lsp-retransmit-interval {
        type rt-types:timer-value-seconds16;
        units "seconds";
        description
         "This leaf defines the interval between
         retransmission of LSP";
      }
      leaf passive {
        type boolean;
        default "false";
```

```
        description
         "This leaf defines if interface is in
          passive mode (ISIS not running,
          but network is advertised).";
      }
      leaf csnp-interval {
        type rt-types:timer-value-seconds16;
        units "seconds";
        default 10;
        description
         "This leaf defines the interval of CSNP
          messages.";
      }
      container hello-padding {
        leaf enable {
          type boolean;
          default "true";
          description
           "Status of Hello-padding activation.
            By default, the implementation shall
            pad HELLOs.";
        }
        description
         "This container handles ISIS hello padding
          configuration.";
      }
      leaf mesh-group-enable {
        type mesh-group-state;
        description
         "Describes the mesh group state of
          the interface.";
      }
      leaf mesh-group {
        when "../mesh-group-enable = 'meshSet'" {
          description
           "Only valid when mesh-group-enable
            equals meshSet";
        }
        type uint8;
        description
         "Describes the mesh group ID of
          the interface.";
      }
      leaf interface-type {
        type interface-type;
        description
         "This leaf defines the type of adjacency
         to be established on the interface.
```

```
         This is affecting the type of hello
         message that would be used.";
      }

      uses admin-control;

      leaf-list tag {
        if-feature prefix-tag;
        type uint32;
        description
         "This leaf defines list of tags associated
          with the interface.";
      }
      leaf-list tag64 {
        if-feature prefix-tag64;
        type uint64;
        description
         "This leaf defines list of 64bits tags
         associated with the interface.";
      }
      leaf node-flag {
          if-feature node-flag;
          type boolean;
          default false;
          description
           "Set prefix as a node
           representative prefix.";
      }
      container hello-authentication {
          uses hello-authentication-cfg;
          container level-1 {
              uses hello-authentication-cfg;
              description "level-1 specific cfg";
          }
          container level-2 {
              uses hello-authentication-cfg;
              description "level-2 specific cfg";
          }
          description "Authentication type
         to be used in hello messages.";
      }
      container hello-interval {
          uses hello-interval-cfg;
          container level-1 {
              uses hello-interval-cfg;
              description "level-1 specific cfg";
          }
          container level-2 {
```

```
            uses hello-interval-cfg;
            description "level-2 specific cfg";
        }
        description "Interval between
         hello messages.";
    }
    container hello-multiplier {
        uses hello-multiplier-cfg;
        container level-1 {
            uses hello-multiplier-cfg;
            description "level-1 specific cfg";
        }
        container level-2 {
            uses hello-multiplier-cfg;
            description "level-2 specific cfg";
        }
        description "Hello multiplier
        configuration.";
    }
    container priority {
        must '../interface-type = "broadcast"' {
            error-message
             "Priority only applies to broadcast
             interfaces.";
            description
             "Check for broadcast interface.";
        }
        uses priority-cfg;
        container level-1 {
            uses priority-cfg;
            description "level-1 specific cfg";
        }
        container level-2 {
            uses priority-cfg;
            description "level-2 specific cfg";
        }
        description "Priority for DIS election.";
    }
    container metric {
        uses metric-cfg;
        container level-1 {
            uses metric-cfg;
            description "level-1 specific cfg";
        }
        container level-2 {
            uses metric-cfg;
            description "level-2 specific cfg";
        }
```

```
          description "Metric configuration.";
      }
      container bfd {
          if-feature bfd;
          leaf enable {
              type boolean;
              default false;
              description "
                  Enables BFD on the interface
              ";
          }
          description
           "BFD configuration.";
      }
      container afs {
          if-feature nlpid-control;
          list af {
            key af;
            leaf af {
              type identityref {
                base rt-types:address-family;
              }
              description
                "Address-family";
            }
            description
             "List of AFs.";
          }
          description
               "Container for address-families";
      }
      container mpls {
        container ldp {
          leaf igp-sync {
              if-feature ldp-igp-sync;
              type boolean;
              description
               "Enables IGP/LDP sync.";
          }
          description
           "LDP protocol related configurations.";
        }
        description
         "Container for MPLS specific configuration
         for ISIS.";
      }
    }
```

```
    grouping isis-if-topologies-cfg {
      description
       "ISIS interface topology cfg.";
      container metric {
          uses metric-cfg;
          container level-1 {
              uses metric-cfg;
              description "level-1 specific cfg";
          }
          container level-2 {
              uses metric-cfg;
              description "level-2 specific cfg";
          }
          description "Metric configuration.";
      }
    }

    grouping system-counters {
     container system-counters {
        list level {
          key level;

          leaf level {
            type level-number;
            description
             "This leaf describes the ISIS 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 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
             occured.";
          }
          leaf id-len-mismatch {
            type uint32;
            description
             "Number of times a PDU is received with
              a different value for ID field length
              from 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 with errors we have
             received.";
          }
          leaf spf-runs {
```

```
            type uint32;
            description
             "Number of times we ran SPF at this level.";
          }
          description
           "List of supported levels.";
        }
        description
         "The container defines a list of counters
          for the IS.";
      }
      description
       "Grouping for system counters.";
    }

    grouping event-counters {
        container event-counters {
            leaf adjacency-changes {
              type uint32;
              description
               "The number of times an adjacency state
                change has occured 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 PPP 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 with
                according max area address field
                differs from that for
                this system has been received on this
                interface.";
            }
            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
             "Provides protocol event counters.";
        }
        description
         "Grouping for event counters";
    }

    grouping packet-counters {
      container packet-counters {
        list level {
          key level;

          leaf level {
            type level-number;
            description
             "This leaf describes the ISIS level.";
          }
          container iih {
```

```
            leaf in {
              type uint32;
              description
               "Received PDUs.";
            }
            leaf out {
              type uint32;
              description
               "Sent PDUs.";
            }
            description
             "The number of IIH PDUs received/sent.";
          }
          container ish {
            leaf in {
              type uint32;
              description
               "Received PDUs.";
            }
            leaf out {
              type uint32;
              description
               "Sent PDUs.";
            }
            description
             "The number of ISH PDUs received/sent.";
          }
          container esh {
            leaf in {
              type uint32;
              description
               "Received PDUs.";
            }
            leaf out {
              type uint32;
              description
               "Sent PDUs.";
            }
            description
             "The number of ESH PDUs received/sent.";
          }
          container lsp {
            leaf in {
              type uint32;
              description
               "Received PDUs.";
            }
            leaf out {
```

```
              type uint32;
              description
               "Sent PDUs.";
            }
            description
             "The number of LSP PDUs received/sent.";
          }
          container psnp {
            leaf in {
              type uint32;
              description
               "Received PDUs.";
            }
            leaf out {
              type uint32;
              description
               "Sent PDUs.";
            }
            description
             "The number of PSNP PDUs received/sent.";
          }
          container csnp {
            leaf in {
              type uint32;
              description
               "Received PDUs.";
            }
            leaf out {
              type uint32;
              description
               "Sent PDUs.";
            }
            description
             "The number of CSNP PDUs received/sent.";
          }
          container unknown {
            leaf in {
              type uint32;
              description
               "Received PDUs.";
            }
            leaf out {
              type uint32;
              description
               "Sent PDUs.";
            }
            description
             "The number of unknown PDUs received/sent.";
```

```
          }
          description
           "List of supported levels.";
        }
        description
         "Provides packet counters per level.";
      }
      description
       "Grouping for packet counters.";
    }

    grouping spf-log {
      container spf-log {
        list event {
          key id;

          leaf id {
            type uint32;
            description
             "This leaf defines the event identifier.
              This is a purely internal value.";
          }
          leaf spf-type {
            type enumeration {
              enum full {
                description
                 "Computation done is a Full SPF.";
              }
              enum route-only {
                description
                 "Computation done is a
                  reachability computation
                  only.";
              }
            }
            description
             "This leaf describes the type of computation
             used.";
          }
          leaf level {
            type level-number;
            description
             "This leaf describes the level affected by the
             the computation.";
          }
          leaf schedule-timestamp {
            type yang:timestamp;
            description
```

```
             "This leaf describes the timestamp
              when the computation was scheduled.";
          }
          leaf start-timestamp {
            type yang:timestamp;
            description
             "This leaf describes the timestamp
              when the computation was started.";
          }
          leaf end-timestamp {
            type yang:timestamp;
            description
             "This leaf describes the timestamp
              when the computation was ended.";
          }
          list trigger-lsp {
            key "lsp";
            leaf lsp {
              type lsp-id;
              description
               "This leaf describes the LSPID
                of the LSP.";
            }
            leaf sequence {
              type uint32;
              description
               "This leaf describes the sequence
                number of the LSP.";
            }
            description
             "This leaf describes list of LSPs
              that triggered the computation.";
          }
          description
           "List of computation events.
           It is used as a wrapping buffer.";
        }

        description
         "This container lists the SPF computation events.";
      }
      description
       "Grouping for spf-log events.";
    }

    grouping lsp-log {
      container lsp-log {
        list event {
```

```
          key id;

          leaf id {
            type uint32;
            description
             "This leaf defines the event identifier.
              This is a purely internal value.";
          }
          leaf level {
            type level-number;
            description
             "This leaf describes the level affected by the
             the computation.";
          }
          container lsp {
            leaf lsp {

              type lsp-id;
              description
               "This leaf describes the LSPID
                of the LSP.";
            }
            leaf sequence {
              type uint32;
              description
               "This leaf describes the sequence
                number of the LSP.";
            }
            description
             "This container describes the received LSP
             , in case of local LSP update the local
             LSP ID is referenced.";
          }

          leaf received-timestamp {
            type yang:timestamp;

            description
             "This leaf describes the timestamp
             when the LSP was received. In case of
             local LSP update, the timestamp refers
             to the local LSP update time.";
          }

          leaf change {
            type identityref {
              base lsp-log-reason;
            }
```

```
            description
             "This leaf describes the type of change
             in the LSP.";
          }

          description
           "List of LSP events.
           It is used as a wrapping buffer.";
        }

        description
         "This container lists the LSP reception events.
         Local LSP modification are also contained in the
         list.";
      }
      description
       "Grouping for LSP log.";
    }

    grouping hostname-db {
      container hostnames {
        list hostname {
          key system-id;
          leaf system-id {
            type system-id;
            description
             "This leaf describes the system-id
             associated with the hostname.";
          }
          leaf hostname {
            type string;
            description
             "This leaf describes the hostname
              associated with the system ID.";
          }
          description
           "List of system-id/hostname associations";
        }
        description
         "This container describes the list
         of binding between system-id and
         hostnames.";
      }
      description
       "Grouping for hostname to systemid mapping database.";
    }

    /* Groupings for the LSDB description */
```

```
    grouping prefix-reachability-attributes {
      description
       "This group defines 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
           "Readvertisement flag.";
      }
      leaf node-flag {
          type boolean;
          description
           "Node flag.";
      }
    }

    grouping prefix-ipv4-source-router-id {
      description
       "This group defines 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
       "This group defines 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.";

      uses prefix-reachability-attributes;
      uses prefix-ipv4-source-router-id;
      uses prefix-ipv6-source-router-id;
    }

    grouping prefix-ipv4-std {
      description
       "This group defines attributes of an
        IPv4 standard prefix.";
      leaf up-down {
        type boolean;
        description
         "This leaf expresses the value of up/down bit.";
      }
      leaf i-e {
        type boolean;
        description
         "This leaf expresses the value of I/E bit.";
      }
      leaf ip-prefix {
        type inet:ipv4-address;
        description
         "This leaf describes the IPv4 prefix";
      }
      leaf prefix-len {
        type uint8;
        description
         "This leaf describes the IPv4 prefix len in bits";
      }
      leaf default-metric {
        type std-metric;
        description
         "This leaf describes the ISIS default metric value";
      }
      container delay-metric {
        leaf metric {
          type std-metric;
          description
           "This leaf describes the ISIS delay metric value";
        }
        leaf supported {
          type boolean;
          default "false";
          description
           "This leaf describes if the metric is supported.";
        }
```

```
        description
         "This container defines the ISIS delay metric.";
      }
      container expense-metric {
        leaf metric {
          type std-metric;
          description
           "This leaf describes the ISIS expense metric value";
        }
        leaf supported {
          type boolean;
          default "false";
          description
           "This leaf describes if the metric is supported.";
        }
        description
         "This container defines the ISIS expense metric.";
      }
      container error-metric {
        leaf metric {
          type std-metric;
          description
           "This leaf describes the ISIS error metric value";
        }
        leaf supported {
          type boolean;
          default "false";
          description
           "This leaf describes if the metric is supported.";
        }

        description
         "This container defines the ISIS error metric.";
      }
    }

    grouping prefix-ipv4-extended {
      description
       "This group defines attributes of an
        IPv4 extended prefix.";
      leaf up-down {
        type boolean;
        description
         "This leaf expresses the value of up/down bit.";
      }
      leaf ip-prefix {
        type inet:ipv4-address;
        description
```

```
         "This leaf describes the IPv4 prefix";
      }
      leaf prefix-len {
        type uint8;
        description
         "This leaf describes the IPv4 prefix len in bits";
      }

      leaf metric {
        type wide-metric;
        description
         "This leaf describes the ISIS metric value";
      }
      leaf-list tag {
        type uint32;
        description
         "This leaf describes a list of tags associated with
         the prefix.";
      }
      leaf-list tag64 {
        type uint64;
        description
         "This leaf describes a list of 64-bit tags associated with
         the prefix.";
      }
      uses prefix-attributes-extension;
    }

    grouping prefix-ipv6-extended {
      description
       "This group defines attributes of an
        IPv6 prefix.";
      leaf up-down {
        type boolean;
        description
         "This leaf expresses the value of up/down bit.";
      }
      leaf ip-prefix {
        type inet:ipv6-address;
        description
         "This leaf describes the IPv6 prefix";
      }
      leaf prefix-len {
        type uint8;
        description
         "This leaf describes the IPv4 prefix len in bits";
      }
      leaf metric {
```

```
        type wide-metric;
        description
         "This leaf describes the ISIS metric value";
      }
      leaf-list tag {
        type uint32;
        description
         "This leaf describes a list of tags associated with
         the prefix.";
      }
      leaf-list tag64 {
        type uint64;
        description
         "This leaf describes a list of 64-bit tags associated with
         the prefix.";
      }
      uses prefix-attributes-extension;
    }

    grouping neighbor-extended {
      description
       "This group defines attributes of an
        ISIS extended neighbor.";
      leaf neighbor-id {
        type system-id;
        description
         "This leaf describes the system-id of the neighbor.";
      }
      leaf metric {
        type wide-metric;
        description
         "This leaf describes the ISIS metric value";
      }
    }

    grouping neighbor {
      description
       "This group defines attributes of an
        ISIS standard neighbor.";
      leaf neighbor-id {
        type system-id;
        description
         "This leaf describes the system-id of the neighbor.";
      }
      leaf i-e {
        type boolean;
        description
         "This leaf expresses the value of I/E bit.";
```

```
      }
      leaf default-metric {
        type std-metric;
        description
         "This leaf describes the ISIS default metric value";
      }
      container delay-metric {
        leaf metric {
          type std-metric;
          description
           "This leaf describes the ISIS delay metric value";
        }
        leaf supported {
          type boolean;
          default "false";
          description
           "This leaf describes if the metric is supported.";
        }
        description
         "This container defines the ISIS delay metric.";
      }
      container expense-metric {
        leaf metric {
          type std-metric;
          description
           "This leaf describes the ISIS delay expense value";
        }
        leaf supported {
          type boolean;
          default "false";
          description
           "This leaf describes if the metric is supported.";
        }
        description
         "This container defines the ISIS expense metric.";
      }
      container error-metric {
        leaf metric {
          type std-metric;
          description
           "This leaf describes the ISIS error metric value";
        }
        leaf supported {
          type boolean;
          default "false";
          description
           "This leaf describes if the metric is supported.";
        }
```

```
        description
         "This container defines the ISIS error metric.";
      }
    }

    grouping lsp-entry {
     description
       "This group defines attributes of an
        ISIS LSP database entry.";

      leaf decoded-completed {
        type boolean;
        description
          "The IS-IS body is fully decoded.";
      }
      leaf raw-data {
        type yang:hex-string;
        description
          "The complete LSP in network byte
           order hexadecimal as received or originated.";
      }
      leaf lsp-id {
        type lsp-id;
        description
         "This leaf describes the LSP ID of the LSP.";
      }
      leaf checksum {
        type uint16;
        description
         "This leaf describes the checksum of the LSP.";
      }
      leaf remaining-lifetime {
        type uint16;
        units "seconds";
        description
         "This leaf describes the remaining lifetime
          in seconds before the LSP expiration.";
      }
      leaf sequence {
        type uint32;
        description
         "This leaf describes the sequence number of the LSP.";
      }
      leaf attributes {
        type bits {
          bit PARTITIONNED {
            description
             "If set, the originator supports partition
```

```
              repair.";
          }
          bit ATTACHED-ERROR {
            description
             "If set, the originator is attached to
              another area using the referred metric.";
          }
          bit ATTACHED-EXPENSE {
            description
             "If set, the originator is attached to
              another area using the referred metric.";
          }
          bit ATTACHED-DELAY {
            description
             "If set, the originator is attached to
              another area using the referred metric.";
          }
          bit ATTACHED-DEFAULT {
            description
             "If set, the originator is attached to
              another area using the referred metric.";
          }
          bit OVERLOAD {
            description
             "If set, the originator is overloaded,
              and must be avoided in path calculation.";
          }
        }
        description
         "This leaf describes attributes of the LSP.";
      }

      leaf-list ipv4-addresses {
        type inet:ipv4-address;
        description
          "This leaf describes the IPv4 addresses of the node.
          ISIS reference is TLV 132.";
      }

      leaf-list ipv6-addresses {
        type inet:ipv6-address;
        description
         "This leaf describes the IPv6 interface
         addresses of the node.
         ISIS reference is TLV 232.";
      }

      leaf ipv4-te-routerid {
```

```
        type inet:ipv4-address;
        description
          "This leaf describes the IPv4 Traffic Engineering
          router ID of the node.
          ISIS reference is TLV 134.";
      }

      leaf ipv6-te-routerid {
        type inet:ipv6-address;
        description
         "This leaf describes the IPv6 Traffic Engineering
          router ID of the node.
          ISIS reference is TLV 140.";
      }

      leaf-list protocol-supported {
        type uint8;
        description
         "This leaf describes the list of
         supported protocols.
         ISIS reference is TLV 129.";
      }

      leaf dynamic-hostname {
        type string;
        description
         "This leaf describes the name of the node.
         ISIS reference is TLV 137.";
      }

      container authentication {
        leaf authentication-type {
          type string;
          description
           "This leaf describes the authentication type
            to be used.";
        }
        leaf authentication-key {
          type string;
          description
           "This leaf describes the authentication key
            to be used. For security reason, the
            authentication key MUST NOT be presented
            in plaintext format. Authors recommends
            to use MD5 hash to present the authentication-key.";
        }
        description "This container describes authentication
        information of the node. ISIS reference is TLV 10.";
```

```
      }

      container mt-entries {
        list topology {

            leaf MT-ID {
              type uint16 {
                range "0 .. 4095";
              }
              description
               "This leaf defines the identifier
                of a topology.";
            }

            leaf attributes {
              type bits {
                bit OVERLOAD {
                  description
                   "If set, the originator is overloaded,
                    and must be avoided in path
                    calculation.";
                }
                bit ATTACHED {
                  description
                   "If set, the originator is attached to
                    another area using the referred metric.";
                }
              }
              description
               "This leaf describes attributes of the LSP
                for the associated topology.";
            }
            description
             "List of topologies supported.";
        }
        description
         "This container describes the topology supported.
         ISIS reference is TLV 229.";
      }

      list router-capabilities {
        leaf flags {
            type bits {
              bit flooding {
                position 0;
                description
                 "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.";
            }
            bit down {
              position 1;
              description
              "When the IS-IS Router CAPABILITY TLV is
              leaked from level-2 to level-1, 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.
              ";
            }
          }
          description
           "Flags associated with router capability.";
        }
        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 tags.";
        }

        leaf binary {
          type binary;
          description
           "This leaf describes the capability of the node.
           Format is binary according to the protocol encoding.";
        }
        description
         "This container describes the capabilities of the node.
         This container may be extended with detailed
         information.
         ISIS reference is TLV 242.";
      }
```

```
      container is-neighbor {
        list neighbor {
          uses neighbor;
          description
           "List of neighbors.";
        }
        description
         "This leaf describes list of ISIS neighbors.
         ISIS reference is TLV 2.";
      }

      container extended-is-neighbor {
        list neighbor {
          uses neighbor-extended;
          description
           "List of neighbors.";
        }
        description
         "This container describes list of ISIS extended
         neighbors.
         ISIS reference is TLV 22.";
      }

      container ipv4-internal-reachability {
        list prefixes {
          uses prefix-ipv4-std;
          description
           "List of prefixes.";
        }
        description
         "This container describes list of IPv4 internal
         reachability information.
         ISIS reference is TLV 128.";
      }

      container ipv4-external-reachability {
        list prefixes {
          uses prefix-ipv4-std;
          description
           "List of prefixes.";
        }
        description
         "This container describes list of IPv4 external
         reachability information.
         ISIS reference is TLV 130.";
      }

      container extended-ipv4-reachability {
```

```
        list prefixes {
          uses prefix-ipv4-extended;
          description
           "List of prefixes.";
        }
        description
         "This container describes list of IPv4 extended
         reachability information.
         ISIS reference is TLV 135.";
      }

      container mt-is-neighbor {
        list neighbor {
          leaf MT-ID {
            type uint16 {
              range "0 .. 4095";
            }
            description
             "This leaf defines the identifier
              of a topology.";
          }
          uses neighbor-extended;
          description
           "List of neighbors.";
        }
        description
         "This container describes list of ISIS multi-topology
          neighbors.
          ISIS reference is TLV 223.";
      }

      container mt-extended-ipv4-reachability {
        list prefixes {
          leaf MT-ID {
            type uint16 {
              range "0 .. 4095";
            }
            description
             "This leaf defines the identifier
              of a topology.";
          }
          uses prefix-ipv4-extended;
          description
           "List of prefixes.";

        }
        description
         "This container describes list of IPv4
```

```
         reachability information in multi-topology
         environment.
         ISIS reference is TLV 235.";
      }

      container mt-ipv6-reachability {
        list prefixes {
          leaf MT-ID {
            type uint16 {
              range "0 .. 4095";
            }
            description
             "This leaf defines the identifier
              of a topology.";
          }
          uses prefix-ipv6-extended;
          description
           "List of prefixes.";
        }
        description
         "This container describes list of IPv6
         reachability information in multi-topology
         environment.
         ISIS reference is TLV 237.";
      }

      container ipv6-reachability {
        list prefixes {
          uses prefix-ipv6-extended;
          description
           "List of prefixes.";
        }
        description
         "This container describes list of IPv6
         reachability information.
         ISIS reference is TLV 236.";
      }
    }

    grouping lsdb {
      container database {
        list level-db {
          key level;

          leaf level {
            type level-number;
            description
             "Current level number";
```

```
          }
          list lsp {
            key lsp-id;
            uses lsp-entry;
            description
             "List of LSPs in LSDB.";
          }
          description
           "This container describes the list of LSPs
           in the level x database.";
        }
        description
         "This container describes ISIS Link State
          databases.";
      }
      description
       "Grouping for LSDB description.";
    }




    /* Augmentations */

    augment "/rt:routing-state/"
      +"rt:ribs/rt:rib/rt:routes/rt:route" {
      when "rt:source-protocol = 'isis:isis'" {
        description "ISIS-specific route attributes.";
      }
      uses route-content;
      description
       "This augments route object in RIB with ISIS-specific
        attributes.";
    }


    augment "/if:interfaces/if:interface" {
      leaf clns-mtu {
        type uint16;
        description
         "Defines CLNS MTU of the interface.";
      }
      description "ISO interface config.";
    }

    augment "/rt:routing/rt:control-plane-protocols/"
        +"rt:control-plane-protocol" {
      when "rt:type = 'isis:isis'" {
```

```
        description
         "This augment is only valid when routing protocol
          instance type is isis.";
      }
      description
       "This augments a routing protocol instance with ISIS
        specific parameters.";
      container isis {
          must "count(area-address) > 0" {
            error-message "At least one area-address
            must be configured.";
            description
             "Enforce configuration of at least one area.";
          }

          uses isis-global-cfg;

          container fast-reroute {
           if-feature fast-reroute;
           uses fast-reroute-global-cfg;
           description
                "IPFRR.";
          }
          container spf-control {
              container ietf-spf-delay {
                  if-feature ietf-spf-delay;
                  uses ietf-spf-delay-cfg;
                  description
                   "IETF spf delay algorithm configuration.";
              }
              description
               "Container for all SPF computation related
               operations.";
          }
          container topologies {
              if-feature multi-topology;
              list topology {
                key "name";
                leaf enable {
                  type boolean;
                  description
                   "Control enabling of topologies";
                }
                leaf name {
                  type leafref {
                     path "../../../../../../rt:ribs/rt:rib/rt:name";
                   }
```

```
                  description "RIB";
                }

                uses isis-global-topologies-cfg;

                description
                  "List of topologies";
              }
              description
               "Container for multi-topology";
            }
            container interfaces {
              list interface {
                key "name";
                leaf name {
                  type if:interface-ref;

                  description
                   "Reference to the interface within
                    the routing-instance.";
              }
              uses isis-if-cfg;
              container fast-reroute {
                if-feature fast-reroute;
                uses fast-reroute-if-cfg;
                description
                  "IPFRR.";
              }
              container topologies {
                  if-feature multi-topology;
                  list topology {
                      key name;

                      leaf name {
                        type leafref {
                            path "../../../../../../../../"+
                            "rt:ribs/rt:rib/rt:name";
                        }

                        description
                         "Name of RIB.";
                      }
                      uses isis-if-topologies-cfg;
                      description
                       "List of topologies.";
                  }
                description
                 "Container for multi-topology";
```

```
            }
            description
              "List of ISIS interfaces.";
            }
            description
             "This container defines ISIS interface specific
             configuration objects.";
         }

         description
          "This container defines ISIS specific configuration
           objects.";
      }
    }

    augment "/rt:routing-state/"
      +"rt:control-plane-protocols/rt:control-plane-protocol" {
      when "rt:type = 'isis:isis'" {
        description
         "This augment is only valid when routing protocol
          instance type is isis.";
      }
      description
       "This augments routing protocol instance states with ISIS
        specific parameters.";

      container isis {
        config false;
          uses isis-global-cfg;
          container fast-reroute {
            if-feature fast-reroute;
            uses fast-reroute-global-cfg;
            uses fast-reroute-global-state;
            description
             "IPFRR states.";
          }
          container spf-control {
              container ietf-spf-delay {
                  if-feature ietf-spf-delay;
                  uses ietf-spf-delay-cfg;
                  uses ietf-spf-delay-state;
                  description
                   "IETF spf delay algorithm configuration.";
              }
              description
               "Container for all SPF computation related
               operations.";
          }
```

```
          list topologies {
            key name;

            leaf name {
              type leafref {
                 path "../../../../../"
                 +"rt:ribs/rt:rib/rt:name";
               }

              description
                "Name of RIB.";
            }
            uses local-rib;
            description
             "List of topologies.";
          }
          uses local-rib;
          uses system-counters;

          container interfaces {
            list interface {
              key interface;

              leaf interface {
                type string;
                description
                 "This leaf describes the name
                 of the interface.";
              }
              uses isis-if-cfg;
              container fast-reroute {
               if-feature fast-reroute;
               uses fast-reroute-if-cfg;
               description
                "IPFRR.";
              }
              list topologies {
                key name;

                leaf name {
                  type leafref {
                      path "../../../../../"
                      +"../../rt:ribs/rt:rib/rt:name";
                  }
                  description
                    "Name of RIB.";
                }
```

```
                uses isis-if-topologies-cfg;
                uses adjacency-state;

                description
                 "List of topologies.";
              }

              uses adjacency-state;
              uses event-counters;
              uses packet-counters;

              description
               "List of interfaces.";
            }
            description
             "The container defines operational parameters
              of interfaces.";
          }

          uses spf-log;
          uses lsp-log;
          uses hostname-db;
          uses lsdb;

        description
           "This container defines various ISIS states objects.";
      }
    }

    /* RPC methods */

    rpc clear-adjacency {
      description
       "This RPC request clears a particular
        set of ISIS adjacencies. If the operation
        fails for ISIS internal reason, then
        error-tag and error-app-tag should be set
        to a meaningful value.";
      input {

        leaf routing-protocol-instance-name {
          type instance-state-ref;
          mandatory "true";
          description
           "Name of the ISIS protocol instance whose ISIS
            information is being queried.

            If the ISIS instance with name equal to the
```

```
            value of this parameter doesn't exist, then this
            operation SHALL fail with error-tag 'data-missing'
            and error-app-tag
            'routing-protocol-instance-not-found'.";
        }
        leaf level {
          type level;
          description
           "ISIS level of the adjacency to be cleared.
           If ISIS level is level-1-2, both level 1 and level 2
           adjacencies would be cleared.

           If the value provided is different from the one
           authorized in the enum type, then this
            operation SHALL fail with error-tag 'data-missing'
            and error-app-tag
            'bad-isis-level'.
           ";
        }
        leaf interface {
          type string;
          description
           "Name of the ISIS interface.

            If the ISIS interface with name equal to the
            value of this parameter doesn't exist, then this
            operation SHALL fail with error-tag 'data-missing'
            and error-app-tag
            'isis-interface-not-found'.";
        }
      }
    }

    rpc clear-database {
      description
       "This RPC request clears a particular
        ISIS database. If the operation
        fails for ISIS internal reason, then
        error-tag and error-app-tag should be set
        to a meaningful value.";
      input {
        leaf routing-protocol-instance-name {
          type instance-state-ref;
          mandatory "true";
          description
           "Name of the ISIS protocol instance whose ISIS
            information is being queried.
```

```
            If the ISIS instance with name equal to the
            value of this parameter doesn't exist, then this
            operation SHALL fail with error-tag 'data-missing'
            and error-app-tag
            'routing-protocol-instance-not-found'.";
        }
        leaf level {
          type level;
          description
           "ISIS level of the adjacency to be cleared.
           If ISIS level is level-1-2, both level 1 and level 2
           adjacencies would be cleared.

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


    /* Notifications */

    notification database-overload {
      uses notification-instance-hdr;

      leaf overload {
        type enumeration {
          enum "off" {
            description
             "The system has left overload condition.";
          }
          enum "on" {
            description
             "The system is in overload condition.";
          }

        }
        description
           "Describes the new overload state of the instance.";
      }
      description
        "This notification is sent when an ISIS instance
         overload condition changes.";
    }
```

```
    notification lsp-too-large {
      uses notification-instance-hdr;
      uses notification-interface-hdr;

      leaf pdu-size {
        type uint32;
        description
         "Size of the PDU";
      }
      leaf lsp-id {
        type lsp-id;
        description
         "LSP ID.";
      }
      description
        "This notification is sent when we attempt
         to propagate an LSP that is larger than the
         dataLinkBlockSize for the circuit.
         The notification generation must be throttled
         with at least a 5 second gap.
         ";
    }

    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 we find
         that 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
         "Size of the ID length in the received PDU";
      }
      leaf raw-pdu {
        type binary;
        description
         "Received raw PDU.";
      }
      description
        "This notification is sent when we receive a PDU
        with a different value for the System ID length.
        The notification generation must be throttled
        with at least a 5 second gap.
         ";
    }

    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 we receive a PDU
        with a different value for the Maximum Area Addresses.
        The notification generation must be throttled
        with at least a 5 second gap.
         ";
    }

    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 reissue
        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 notification generation must be throttled with
        at least a 5 second gap.
         ";
    }

    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 with the wrong authentication
        information.
        The notification generation must be throttled with
        at least a 5 second gap.
         ";
    }

    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 notification generation must be throttled with at least
         a 5 second gap.
         ";
    }
```

```
    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 notification generation must be throttled with at least
         a 5 second gap.
         ";
    }

    notification rejected-adjacency {
      uses notification-instance-hdr;
      uses notification-interface-hdr;
      leaf raw-pdu {
        type binary;
        description
         "Received raw PDU.";
      }
      leaf reason {
        type string;
        description
         "The system may provide a reason to reject the
          adjacency. If the reason is not available,
          the system use an empty string.";
      }
      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 notification generation must be throttled with at least
         a 5 second gap.
         ";
    }


    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
         "The 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
        protocol supported.
        The notification generation must be throttled with at least
         a 5 second gap.
         ";
    }

    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 suspicious 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 a  LSP with a parse error.
```

```
        The notification generation must be throttled with at least
         a 5 second gap.
         ";
    }

    notification adjacency-state-change {
      uses notification-instance-hdr;
      uses notification-interface-hdr;
      leaf neighbor {
        type string;
        description
         "Describes the name of the neighbor. If the
         name of the neighbor is not available, the
         field would be empty.";
      }
      leaf neighbor-system-id {
        type system-id;
        description
         "Describes the system-id of the neighbor.";
      }
      leaf state {
        type adj-state-type;

        description
         "This leaf describes the new state of the
          ISIS adjacency.";
      }
      leaf reason {
        type string;
        description
         "If the adjacency is going to DOWN,
         this leaf provides a reason for the adjacency
         going down. The reason is provided as a text.
         If the adjacency is going to UP, no reason is
         provided.";
      }
      description
        "This notification is sent when an ISIS adjacency
         moves to Up state or to 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
         "This leaf describes the timestamp
         when the LSP was received. ";
      }
      leaf neighbor-system-id {
        type system-id;
        description
         "Describes the system-id of the neighbor
         that sent the LSP.";
      }
      description
        "This notification is sent when a LSP
        is received.
        The notification generation must be throttled with at least
         a 5 second gap. ";
    }

    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
         "This leaf describes the timestamp
         when our LSP was regenerated. ";
      }
      description
```

```
        "This notification is sent when a LSP
        is regenerated.
        The notification generation must be throttled with at least
         a 5 second gap. ";
    }
  }

  <CODE ENDS>
```

7. Security Considerations

Configuration and state data defined in this document are designed to be accessed via the NETCONF protocol [RFC6241].

As IS-IS is an IGP protocol (critical piece of the network), ensuring stability and security of the protocol is mandatory for the network service.

Authors recommends to implement NETCONF access control model ([RFC6536]) to restrict access to all or part of the configuration to specific users. Access control to RPCs is also critical as RPC allows to clear protocol datastructures that would definitively impact the network service. This kind of RPC needs only to be used in specific cases by well-known experienced users.

Authors consider that all the configuration is considered as sensitive/vulnerable as well as RPCs. But security teams can decide to open some part of the configuration to less experienced users depending on the internal organization, for example:

- User FullWrite: would access to the whole data model. This kind of profile may be restricted to few experienced people.

- User PartialWrite: would only access to configuration part within /isis/interfaces/interface. So this kind of profile is restricted to creation/modification/deletion of interfaces. This profile does not have access to RPC.

- User Read: would only access to the operational states.

Unauthorized access to configuration or RPC may cause high damages to the network service.

The "isis/database" may contain authentication information. As presented in the description of the "/isis/database/level-1/lsp/authentication/authentication-key", the authentication MUST never be displayed in a plaintext format for security reason.

Authors recommend the usage of MD5 to display or return the authentication-key.

Some authentication-key may also be required in the "isis" writable container. When configuring IS-IS using the NETCONF protocol, authors recommends the usage of secure transport of NETCONF using SSH ([RFC6242]).

8. Contributors

Authors would like to thank Kiran Agrahara Sreenivasa, Dean Bogdanovic, Yingzhen Qu, Yi Yang for their major contributions to the draft.

9. Acknowledgements

TBD.

10. IANA Considerations

The IANA is requested to assign two new URIs from the IETF XML registry ([RFC3688]). Authors are suggesting the following URI:

```
URI: urn:ietf:params:xml:ns:yang:ietf-isis
Registrant Contact: IS-IS WG
XML: N/A, the requested URI is an XML namespace
```

This document also requests one new YANG module name in the YANG Module Names registry ([RFC6020]) with the following suggestion:

```
name: ietf-isis
namespace: urn:ietf:params:xml:ns:yang:ietf-isis
prefix: isis
reference: RFC XXXX
```

11. Change log for ietf-isis YANG module

11.1. From version -16 to version -17

- o Cosmetic fixes.
- o Use of rt-types model.

11.2. From version -15 to version -16

- o Alignment with last IETF key chain model.
- o lsp-log "change" leaf moved as an identity.

- o Incremental SPF removed from spf-log types.

11.3. From version -14 to version -15

- o Alignment with OSPF model done:
  - * Added spf-control container with IETF SPF delay algorithm as a feature.
  - * Added graceful-restart options.
  - * Added nsr as a feature.
  - * Removed per topology FRR. Need to be augmented if necessary.
  - * Created an ldp container within mpls.
  - * Renamed igp-ldp-sync to igp-sync.
  - * Added auto-cost container.
  - * Moved reference-bandwidth under auto-cost container.
  - * Added IS-IS local RIB as operational state.
  - * Added decode-completed and raw-data leaves in the LSDB model.
  - * Modified the notification header.

11.4. From version -13 to version -14

- o Segment Routing extensions are now in a separate document.

11.5. From version -12 to version -13

- o Move feature nlpid-control to container rather than list.
- o Rename multi-topology to topologies to align with OSPF.
- o Rename bfd/enabled to bfd/enable for consistency reason.
- o Add support for NSR with a feature.

11.6. From version -09 to version -12

- o Rename node-tag container to node-tags.

11.7. From version -08 to version -09

- o Added container before af list.
- o Added container before topology list.
- o Aligned LFA if per level cfg.
- o Align to draft-ietf-netmod-routing-cfg-23.

11.8. From version -07 to version -08

- o Remove selector from system-id type.
- o Add some default values.
- o Moved lists to containers+groupings for per level configuration.
- o remove routing-instance as per core routing model v21.
- o added BFD leaf (no more BFD protocol model).
- o changed keychain module reference.

11.9. From version -05 to version -07

- o Move Overload config from list to container.
- o Move Overload-max-metric config from list to container.
- o Move preference config from list to container.
- o Add Node flag in config.
- o Removed BFD config => moved to isis-bfd module.
- o Remove call to routing policy model.

11.10. From version -03 to version -05

- o Correct invalid references to previous versions of core routing model.
- o Remove BFD config and replace by groupings from ietf-bfd.
- o Adding routing-policy support through routing-policy model.

11.11. From version -02 to version -03

- o Reviewed config and op state groupings.
- o Add default value to lfa candidate-disabled.
- o Add enable leaf to isis container to reflect admin state.
- o Move to VRF centric only.
- o Segment routing is part os a separate module.

11.12. From version -01 to version -02

- o Adding IPFRR.
- o Adding igp-ldp-sync.
- o Adding segment-routing.
- o Adding instance reference to operational states.
- o Move AF type from string to identity.
- o Updated router-capability in LSDB description.
- o packet counters moved to interface-packet-counters.
- o Added modification information in lsp-log.
- o Removing igp-ldp-sync timer in IS-IS.
- o Defining hierarchy for operational states.
- o Adding clns-mtu.
- o Adding key-chain.

11.13. From version -00 to version -01

- o Interface metric move from af container to interface container.
- o Hello-padding on interface moved to hello-padding-disable with empty type.
- o three-way-handshake removed.
- o route preference changed to a choice.

- o csnp-authentication/psnp-authentication merged to authentication container.
- o lsp-gen-interval-exp-delay removed.
- o Added overload-max-metric feature.
- o overload-max-metric is in a separate container.
- o Change hello-padding to container.
- o Change bfd to container.
- o Make BFD a feature.
- o Create mpls-te container and put router-id inside.
- o Remove GR helper disable and timers.

12. Normative References

[I-D.ietf-netmod-routing-cfg]
Lhotka, L. and A. Lindem, "A YANG Data Model for Routing Management", draft-ietf-netmod-routing-cfg-25 (work in progress), November 2016.

[I-D.ietf-rtgwg-yang-key-chain]
Lindem, A., Qu, Y., Yeung, D., Chen, I., and Z. Zhang, "Routing Key Chain YANG Data Model", draft-ietf-rtgwg-yang-key-chain-17 (work in progress), March 2017.

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

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

[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, <http://www.rfc-editor.org/info/rfc5120>.

[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, <http://www.rfc-editor.org/info/rfc5286>.

[RFC5443] Jork, M., Atlas, A., and L. Fang, "LDP IGP Synchronization", RFC 5443, DOI 10.17487/RFC5443, March 2009, <http://www.rfc-editor.org/info/rfc5443>.

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

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

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

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

[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, <http://www.rfc-editor.org/info/rfc7490>.

Appendix A. Example of IS-IS configuration in XML

This section gives an example of configuration of an IS-IS instance on a device. The example is written in XML.

```
<?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>1.1.1.1</router-id>
      <description/>
      <interfaces>
        <interface>
          <name>Loopback0</name>
        </interface>
```

```
          <interface>
            <name>Eth1</name>
          </interface>
        </interfaces>
        <control-plane-protocols>
          <control-plane-protocol>
            <name>ISIS</name>
            <description/>
            <type>isis:isis</type>
            <isis xmlns="urn:ietf:params:xml:ns:yang:ietf-isis">
                <enable>true</enable>
                <level-type>level-2</level-type>
                <system-id>87FC.FCDF.4432</system-id>
                <area-address>49.0001</area-address>
                <mpls-te>
                  <ipv4-router-id>1.1.1.1</ipv4-router-id>
                </mpls-te>
                <lsp-lifetime>65535</lsp-lifetime>
                <lsp-refresh>65000</lsp-refresh>
                <metric-type>
                  <value>wide</value>
                </metric-type>
                <default-metric>
                  <value>111111</value>
                </default-metric>
                <afs>
                  <af>
                      <af>ipv4-unicast</af>
                      <enabled>true</enabled>
                  </af>
                </afs>
                <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/>
        <link-up-down-trap-enable/>
        <ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
          <mtu/>
          <address>
            <ip>1.1.1.1</ip>
            <prefix-length>32</prefix-length>
          </address>
        </ipv4>

      </interface>
      <interface>
        <name>Eth1</name>
        <description/>
        <type/>
        <link-up-down-trap-enable/>
        <ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
          <mtu/>
          <address>
            <ip>10.0.0.1</ip>
            <prefix-length>30</prefix-length>
          </address>
        </ipv4>

      </interface>
    </interfaces>
  </data>
```


Authors' Addresses

Stephane Litkowski
Orange

Email: stephane.litkowski@orange.com

Derek Yeung
Arrcus, Inc

Email: derek@arrcus.com

Acee Lindem
Cisco Systems

Email: acee@cisco.com

Jeffrey Zhang
Juniper Networks

Email: zzhang@juniper.net

Ladislav Lhotka
CZ.NIC

Email: lhotka@nic.cz