IS-IS WG
Internet-Draft
Intended status: Standards Track
Expires: March 19, 2018

S. Hegde
C. Bowers
Juniper Networks
P. Mattes
M. Nanduri
S. Giacalone
Microsoft
I. Mohammad
Arista Networks

September 15, 2017

Advertising TE protocols in IS-IS
draft-hegde-isis-advertising-te-protocols-03

Abstract

This document defines a mechanism to indicate which traffic engineering protocols are enabled on a link in IS-IS. It does so by introducing a new traffic-engineering protocol sub-TLV for TLV-22. This document also describes mechanisms to address backward compatibility issues for implementations that have not yet been upgraded to software that understands this new sub-TLV.

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 RFC 2119 [RFC2119].

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

IS-IS extensions for traffic engineering are specified in [RFC5305]. [RFC5305] defines several link attributes such as administrative group, maximum link bandwidth, and shared risk link groups (SRLGs) which can be used by traffic engineering applications. Additional link attributes for traffic engineering have subsequently been defined in other documents as well. Most recently [RFC7810] defined link attributes for delay, loss, and measured bandwidth utilization.

The primary consumers of these traffic engineering link attributes have been RSVP-based applications that use the advertised link attributes to compute paths which will subsequently be signalled using RSVP-TE. However, these traffic engineering link attributes have also been used by other applications, such as IP/LDP fast-reroute using loop-free alternates as described in [RFC7916]. In the future, it is likely that traffic engineering applications based on Segment Routing [I-D.ietf-spring-segment-routing] will also use these link attributes.

Existing IS-IS standards do not provide a mechanism to explicitly indicate whether or not RSVP has been enabled on a link. Instead, different RSVP-TE implementations have used the presence of certain traffic engineering sub-TLVs in IS-IS to infer that RSVP signalling is enabled on a given link. A study was conducted with various vendor implementations to determine which traffic engineering sub-TLVs cause an implementation to infer that RSVP signalling is enabled on a link. The results are shown in Figure 1.

TLV/ sub-TLV	Sub-TLV name	Implementation		
		X	Y	Z
22	Extended IS Reachability TLV	N	N	N
22/3	Administrative group (color)	N	Y	Y
22/4	Link Local/Remote ID	N	N	N
22/6	IPV4 Interface Address	N	N	N
22/8	IPV4 Neighbor Address	N	N	N
22/9	Max Link Bandwidth	N	Y	Y
22/10	Max Reservable Link Bandwidth	N	Y	Y
22/11	Unreserved Bandwidth	Y	Y	Y
22/14	Extended Admin Group	N	Y	N
22/18	TE Default Metric	N	N	N
22/20	Link Protection Type	N	Y	Y
22/21	Interface Switching Capability	N	Y	Y
22/22	TE Bandwidth Constraints	N	Y	Y
22/33-39	TE Metric Extensions(RFC7180)	N	N	N
138	SRLG TLV	N	Y	Y

Figure 1: Traffic engineering Sub-TLVs that cause implementation X, Y, or Z to infer that RSVP signalling is enabled on a link

The study indicates that the different implementations use the presence of different sub-TLVs under TLV 22 (or the presence of TLV 138) to infer that RSVP signalling is enabled on a link. It is possible that other implementations may use other sub-TLVs to infer that RSVP is enabled on a link.

This document defines a standard way to indicate whether or not RSVP, segment routing, or another future protocol is enabled on a link. In this way, implementations will not have to infer whether or not RSVP is enabled based on the presence of different sub-TLVs, but can use the explicit indication. When network operators want to use a non-RSVP traffic engineering application (such as IP/LDP FRR or segment routing), they will be able to advertise traffic engineering sub-TLVs and explicitly indicate what traffic engineering protocols are enabled on a link.

2. Goals

1. The solution should allow the TE protocol enabled on a link to be communicated unambiguously.

2. The solution should decouple the advertisement of which TE protocols are enabled on a link from the advertisement of other TE attributes.

3. The solution should be backward compatible so that nodes that do not understand the new advertisement do not cause issues for existing RSVP deployments.

4. The solution should be extensible for new protocols.

5. The solution should try to limit any increases to the quantity and size of link state advertisements.

2.1. Explicit and unambiguous indication of TE protocol

Communicating unambiguously which TE protocol is enabled on a link is important to be able to share this information with other consumers through other protocols, aside from just the IGP. For example, for a network running both RSVP-TE and SR, it will be useful to communicate which TE protocols are enabled on which links via BGP-LS [RFC7752] to a central controller. Typically, BGP-LS relies on the IGP to distribute IGP topology and traffic engineering information so that only a few BGP-LS sessions with the central controller are needed. In order for a router running a BGP-LS session to a central controller to correctly communicate what TE protocols are enabled on the links in the IGP domain, that information first needs to be communicated unambiguously within the IGP itself. As Figure 1 illustrates, that is currently not the case.

3. Solution

3.1. Traffic-engineering protocol sub-TLV

A new Traffic-engineering protocol sub-TLV is added in the TLV 22 [RFC5305] or TLV 222 to indicate the protocols enabled on the link. The sub-TLV has flags in the value field to indicate the protocol enabled on the link. The length field is variable to allow the flags field to grow for future requirements.

```
 Type  : TBD suggested value 40
 Length: Variable
 Value :
   0                   1                   2                   3
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                             Flags                             |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: Traffic-Engineering Protocol sub-TLV

Type : TBA (suggested value 40)

Length: variable (in bytes)

Value: The value field consists of bits indicating the protocols enabled on the link. This document defines the two protocol values below.

Value	Protocol Name
0x01	RSVP
0x02	Segment Routing

Figure 3: Flags for the protocols

The RSVP flag is set to one to indicate that RSVP-TE is enabled on a link. The RSVP flag is set to zero to indicate that RSVP-TE is not enabled on a link.

The Segment Routing flag is set to one to indicate that Segment Routing is enabled on a link. The Segment Routing flag is set to zero to indicate that Segment Routing is not enabled on a link.

All undefined flags MUST be set to zero on transmit and ignored on receipt.

An implementation that supports the TE protocol sub-TLV and sends TLV 22 MUST advertise the TE protocol sub-TLV in TLV 22 for that link, even when both the RSVP and SR flags are set to zero. In other words, whenever the TE protocol sub-TLV is supported, it MUST be sent, even if no TE protocols are enabled on the link. This allows a receiving router to determine whether or not the sending router is capable of sending the TE protocol sub-TLV.

A router supporting the TE protocol sub-TLV which receives an advertisement for a link containing TLV 22 with the TE protocol sub-TLV present SHOULD respect the values of the flags in the TE protocol sub-TLV. The receiving router SHOULD only consider links with a given TE protocol enabled for inclusion in a path using that TE protocol. Conversely, links for which the TE protocol sub-TLV is present, but for which the TE protocol flag is not set to one, SHOULD NOT be included in any TE CSPF computations on the receiving router for the protocol in question.

The ability for a receiving router to determine whether or not the sending router is capable of sending the TE protocol sub-TLV is also used for backward compatibility as described in Section 4.

An implementation that supports the TE protocol sub-TLV SHOULD be able to advertise TE sub-TLVs without enabling RSVP-TE signalling on the link.

3.2. Segment Routing flag considerations

The Segment Routing (SR) architecture assumes that the SR topology is congruent with the IGP topology. The path described by a prefix segment is computed using the SPF algorithm applied to the IGP topology, which is the same as the SR topology. Therefore, the presence or absence of the Segment Routing flag MUST NOT be interpreted as modifying the SR topology, which is always congruent with the IGP topology.

It is however useful for a centralized application (or an ingress router) to know whether or not it should expect to be able to forward traffic over a given link using labels distributed via SR. If a link is advertised with the TE protocol sub-TLV and the SR flag set to zero, then a centralized application can assume that traffic sent with a prefix segment whose path crosses that link is unlikely to be forwarded across that link. With this information, a centralized application can decide to use a different path for that traffic by using a different label stack.

4. Backward compatibility

Routers running older software that do not understand the new Traffic-Engineering protocol sub-TLV will continue to interpret the presence of some sub-TLVs in TLV 22 or the presence of TLV 138 as meaning that RSVP is enabled a link. A network operator may not want to or be able to upgrade all routers in the domain at the same time. There are two backward compatibility scenarios to consider depending on whether the router that doesn’t understand the new TE protocol sub-TLV is an RSVP-TE ingress router or an RSVP-TE transit router.

4.1. Scenario with upgraded RSVP-TE transit router but RSVP-TE ingress router not upgraded

An upgraded RSVP-TE transit router is able to explicitly indicate that RSVP is not enabled on a link by advertising the TE protocol sub-TLV with the RSVP flag set to zero. However, an RSVP-TE ingress router that has not been upgraded to understand the new TE protocol sub-TLV will not understand that RSVP-TE is not enabled on the link, and may include the link on a path computed for RSVP-TE. When the network tries to signal an explicit path LSP using RSVP-TE through that link, it will fail. In order to avoid this scenario, an operator can use the mechanism described below.

For this scenario, the basic idea is to use the existing administrative group link attribute as a means of preventing existing RSVP implementations from using a link. The network operator defines an administrative group to mean that RSVP is not enabled on a link. We call this admin group the RSVP-not-enabled admin group. If the operator needs to advertise a TE sub-TLV (maximum link bandwidth, for example) on a link, but doesn’t want to enable RSVP on that link, then the operator also advertises the RSVP-not-enabled admin group on that link. The operator can then use existing mechanisms to exclude links advertising the RSVP-not-enabled admin group from the constrained shortest path first (CSPF) computation used by RSVP. This will prevent RSVP implementations from attempting to signal RSVP-TE LSPs across links that do not have RSVP enabled. Once the entire network domain is upgraded to understand the TE protocol sub-TLV in this draft, the configuration involving the RSVP-not-enabled admin group is no longer needed for this network.

4.2. Scenario with upgraded RSVP-TE ingress router but RSVP-TE transit router not upgraded

The other scenario to consider is when the RSVP-TE ingress router has been upgraded to understand the TE protocol sub-TLV, but the RSVP-TE transit router has not. In this case, the transit router has not been upgraded, so it is not yet capable of sending the TE protocol sub-TLV. If the transit router has RSVP-TE enabled on a link, we would like for the RSVP-TE ingress router to still be able to use the link for RSVP-TE paths. While it is possible to describe a solution for this scenario that makes use of administrative groups, we describe a simpler solution below.

The solution for this scenario relies on the following observation. If the RSVP-TE ingress router can understand that the transit router is not capable of sending the TE protocol sub-TLV, then it can continue inferring whether or not RSVP-TE is enabled on the transit router links based on the presence of TE sub-TLVs, just as it does today.

To accomplish this, we require an upgraded router to send the TE protocol sub-TLV if it sends TLV 22, even when both the RSVP and SR flags are set to zero. In other words, whenever the TE protocol sub-TLV is supported, it MUST be sent, even if no TE protocols are enabled on the link. see Section 3. This allows the receiving router to interpret the absence of the TE-protocol sub-TLV together with presence of TLV 22 to mean that the sending router has not been upgraded. This allows the upgraded RSVP-TE ingress router to distinguish between transit routers that have been upgraded and those that haven't. When the transit router has been upgraded, then the RSVP-TE ingress router uses the information in the TE protocol sub-TLV. When the transit router has not been upgraded, then RSVP-TE ingress router contines to infer whether or not RSVP-TE is enabled on the transit router links based on the presence of TE sub-TLVs, just as it does today. The solution for this scenario requires no configuration on the part of network operators.

4.3. Need for a long term solution

The use of the adminstrative group link attribute to prevent an RSVP-TE ingress router from computing a path using a given link is an effective short term workaround to allow networks to incrementally upgrade the routers to software that understands the new TE-protocol sub-TLV. One might also consider a long term solution based solely on the use of operator-defined adminstrative groups to communicate the TE protocol enabled on a link. However, we do not consider this workaround to be an effective long term solution because it relies on operator configuration that would have to be maintained in the long term. As discussed in Section 2, continuing to have to infer which TE protocol is enabled on a link also limits our ability to communicate this information unambiguously in an interoperable manner for use by other applications such as central controllers.

5. Security Considerations

This document does not introduce any further security issues other than those discussed in [RFC1195] and [RFC5305].

6. IANA Considerations

This specification updates one IS-IS registry:

The extended IS reachability TLV Registry

i) Traffic-engineering Protocol sub-tlv = Suggested value 40

7. Acknowledgements

The authors thank Alia Atlas, Les Ginsberg, and Peter Psenak for helpful discussions on the topic of this draft.

8. References

8.1. Normative References

[I-D.ietf-spring-segment-routing]
Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", draft-ietf-spring-segment-routing-09 (work in progress), July 2016.

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

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <https://www.rfc-editor.org/info/rfc5305>.

[RFC7810] Previdi, S., Ed., Giacalone, S., Ward, D., Drake, J., and Q. Wu, "IS-IS Traffic Engineering (TE) Metric Extensions", RFC 7810, DOI 10.17487/RFC7810, May 2016, <https://www.rfc-editor.org/info/rfc7810>.

8.2. Informative References

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

[RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <https://www.rfc-editor.org/info/rfc7752>.

[RFC7916] Litkowski, S., Ed., Decraene, B., Filsfils, C., Raza, K., Horneffer, M., and P. Sarkar, "Operational Management of Loop-Free Alternates", RFC 7916, DOI 10.17487/RFC7916, July 2016, <https://www.rfc-editor.org/info/rfc7916>.

Authors' Addresses

Shraddha Hegde
Juniper Networks
Embassy Business Park
Bangalore, KA 560093
India

Email: shraddha@juniper.net

Chris Bowers
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
US

Email: cbowers@juniper.net

Paul Mattes
Microsoft
One Microsoft Way
Redmond, WA 98052
US

Email: pamattes@microsoft.com

Mohan Nanduri
Microsoft
One Microsoft Way
Redmond, WA 98052
US

Email: mnanduri@microsoft.com

Spencer Giacalone
Microsoft
One Microsoft Way
Redmond, WA 98052
US

Email: Spencer.Giacalone@microsoft.com

Imtiyaz Mohammad
Arista Networks
Global Tech Park
Bangalore, KA 560103
India

Email: imtiyaz@arista.com

isis B. Liu, Ed.
Internet-Draft Huawei Technologies
Intended status: Standards Track L. Ginsberg
Expires: November 10, 2017 Cisco Systems
B. Decraene
Orange
I. Farrer
Deutsche Telekom AG
M. Abrahamsson
T-Systems
May 9, 2017

ISIS Auto-Configuration
draft-ietf-isis-auto-conf-05

Abstract

This document specifies IS-IS auto-configuration mechanisms. The key components are IS-IS System ID self-generation, duplication detection and duplication resolution. These mechanisms provide limited IS-IS functions, and so are suitable for networks where plug-and-play configuration is expected.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] when they appear in ALL CAPS. When these words are not in ALL CAPS (such as "should" or "Should"), they have their usual English meanings, and are not to be interpreted as [RFC2119] key words.

Status of This Memo

This Internet-Draft will expire on November 10, 2017.

Table of Contents

1. Introduction

This document specifies mechanisms for IS-IS [RFC1195] [ISO_IEC10589][RFC5308] to be auto-configuring. Such mechanisms could reduce the management burden for configuring a network, especially where plug-and-play device configuration is required.

IS-IS auto-configuration is comprised of the following functions:

1. IS-IS default configuration.

2. IS-IS System ID self-generation.

3. System ID duplication detection and resolution.

4. ISIS TLV utilization (Authentication TLV, metrics in reachability advertisements, and Dynamic Host Name TLV).

This document also defines mechanisms to prevent the unintentional interoperation of auto-configured routers with non-autoconfigured routers. See Section 3.3.

2. Scope

The auto-configuration mechanisms support both IPv4 and IPv6 deployments.

These auto-configuration mechanisms aim to cover simple deployment cases. The following important features are not supported:

- Multiple IS-IS instances.

- Multi-area and level-2 routing.

- Interworking with other routing protocols.

IS-IS auto-configuration is primarily intended for use in small (i.e. 10s of devices) and unmanaged deployments. It allows IS-IS to be used without the need for any configuration by the user. It is not recommended for larger deployments.

3. Protocol Specification

3.1. IS-IS Default Configuration

- IS-IS interfaces MUST be auto-configured to an interface type corresponding to their layer-2 capability. For example, Ethernet interfaces will be auto-configured as broadcast networks and

Point-to-Point Protocol (PPP) interfaces will be auto-configured as Point-to-Point interfaces.

- IS-IS auto-configuration instances MUST be configured as level-1, so that the interfaces operate as level-1 only.

- originatingLSPBufferSize is set to 512.

- MaxAreaAddresses is set to 3

- Extended IS Reachability and IP Reachability TLVs [RFC5305] MUST be used i.e. a router operating in auto configuration mode MUST NOT use any of the following TLVs:

  * IS Neighbors (2)

  * IP Internal Reachability (128)

  * IP External Reachability (130)

  TLVs listed above MUST be ignored on receipt.

3.2. IS-IS NET Generation

In IS-IS, a router (known as an Intermediate System) is identified by a Network Entity Title (NET) which is a type of Network Service Access Point (NSAP). The NET is the address of an instance of the IS-IS protocol running on an Intermediate System (IS).

The auto-configuration mechanism generates the IS-IS NET as the following:

- Area address

  In IS-IS auto-configuration, this field MUST be 13 octets long and set to all 0.

- System ID

  This field follows the area address field, and is 6 octets in length. There are two basic requirements for the System ID generation:

  - As specified by the IS-IS protocol, this field must be unique among all routers in the same area.

  - After its initial generation, the System ID SHOULD remain stable. Changes such as interface enable/disable, interface connect/disconnect, device reboot, firmware update, or configuration changes SHOULD NOT cause the system ID to change. System ID change as part of the System ID collision resolution process MUST be supported. Implementations SHOULD allow the System ID to be cleared by a user initiated system reset.

More specific considerations for System ID generation are described in Section 3.4.5.

3.3. Router-Fingerprint TLV

The Router-Fingerprint TLV is similar to the Router-Hardware-Fingerprint TLV defined in [RFC7503]. However, the TLV defined here includes a flags field to support indicating that the router is in Start-up mode and is operating in auto-configuration mode.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Flags Field  |                                               |
+-+-+-+-+-+-+-+-+       Router Fingerprint (Variable)           .
.                                                               .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type: to be assigned by IANA.
Length: the length of the value field. Must be >= 33.
Flags field (1 octet)

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|S|A| Reserved  |
+-+-+-+-+-+-+-+-+
```

S flag: when set, indicates the router is in "start-up" mode.
A flag: when set, indicates that the router is operating in auto-configuration mode. The purpose of the flag is so that two routers can identify if they are both using auto-configuration. If the A flag setting does not match in hellos then no adjacency should be formed.
Reserved: these bits MUST be set to zero and MUST be ignored by the receiver.

Router Fingerprint: 32 or more octets.

More specific considerations for Router-Fingerprint are described in Section 3.4.5.

Router Fingerprint TLV MUST be included in Intermediate System to Intermediate System Hellos (IIHs) originated by a router operating in auto-configuration mode. An auto-configuration mode router MUST ignore IIHs that don’t contain the Router Fingerprint TLV.

Router Fingerprint TLV MUST be included in Link State PDU (LSP) #0 originated by a router operating in auto-configuration mode. If an LSP #0 which does NOT contain a Router Fingerprint TLV is received by a Router operating in auto-configuration mode the LSP is flooded as normal, but the entire LSP set originated by the sending router MUST be ignored when running the Decision process.

The router fingerprint TLV MUST NOT be included in an LSP with a non-zero number and when received MUST be ignored.

3.4. Protocol Operation

This section describes the operation of a router supporting auto-configuration mode.

3.4.1. Start-Up mode

When a router starts operation in auto-configuration mode, both the S and A bits MUST be set in the Router Fingerprint TLV included in both hellos and LSP #0. During this mode only LSP #0 is generated and IS or IP/IPv6 reachability TLVs MUST NOT be included in LSP #0. A router remains in Start-up mode for a minimum period of time (recommended to be 1 minute). This time should be sufficient to bring up adjacencies to all expected neighbors. A router leaves Start-up mode once the minimum time has elapsed and full LSP database synchronization is achieved with all neighbors in the UP state.

When a router exits startup-mode it clears the S bit in Router Fingerprint TLVs it sends in hellos and LSP#0. The router MAY now advertise IS neighbor and IP/IPv6 prefix reachability in its LSPs and MAY generate LSPs with a non-zero number.

The purpose of Start-up Mode is to minimize the occurrence of System ID changes for a router once it has become fully operational. Any System ID change during Start-up mode will have minimal impact on a running network because while in Start-up mode the router is not yet being used for forwarding traffic.

3.4.2. Adjacency Formation

Routers operating in auto-configuration mode MUST NOT form adjacencies with routers which are NOT operating in auto-configuration mode. The presence of the Router Fingerprint TLV with the A bit set indicates the router is operating in auto-configuration mode.

NOTE: The use of the special area address of all 0's makes it unlikely that a router which is not operating in auto-configuration mode will be in the same area as a router operating in auto-configuration mode. However, the check for the Router Fingerprint TLV with A bit set provides additional protection.

3.4.3. IS-IS System ID Duplication Detection

The System ID of each node MUST be unique. As described in Section 3.4.5, the System ID is generated based on entropies (e.g. MAC address) which are generally expected to be unique. However, since there may be limitations to the available entropies, there is still the possibility of System ID duplication. This section defines how IS-IS detects and resolves System ID duplication. Duplicate System ID may occur between neighbors or between routers in the same area which are not neighbors.

Duplicate System ID with a neighbor is detected when the System ID received in an IIH is identical to the local System ID and the Router-Fingerprint in the received Router-Fingerprint TLV does NOT match the locally generated Router-Fingerprint.

Duplicate System ID with a non-neighbor is detected when an LSP #0 is received, the System ID of the originator is identical to the local System ID, and the Router-Fingerprint in the Router-Fingerprint TLV does NOT match the locally generated Router-Fingerprint.

3.4.4. Duplicate System ID Resolution Procedures

When duplicate System ID is detected one of the systems MUST assign itself a different System ID and perform a protocol restart. The resolution procedure attempts to minimize disruption to a running network by choosing a router which is in Start-up mode to be restarted whenever possible.

The contents of the Router-Fingerprint TLVs for the two routers with duplicate System IDs are compared.

If one TLV has the S bit set (router is in Start-up mode) and one TLV has the S bit clear (router is NOT in Start-up mode) the router in Start-up mode MUST generate a new System ID and restart the protocol.

If both TLVs have the S bit set (both routers are in Start-up mode) or both TLVs have the S bit clear (neither router is in Start-up mode) then the router with numerically smaller Router-Fingerprint MUST generate a new System ID and restart the protocol.

Fingerprint comparison is performed octet by octet starting from the first received octet until a difference is detected. If the fingerprints have different lengths and all octets up to the shortest length are identical then the fingerprint with smaller length is considered smaller.

If the fingerprints are identical in both content and length (and state of the S bit is identical) and the duplication is detected in hellos then the both routers MUST generate a new System ID and restart the protocol.

If fingerprints are identical in both content and length and the duplication is detected in LSP #0 then the procedures defined in Section 3.4.6 MUST be followed.

3.4.5. System ID and Router-Fingerprint Generation Considerations

As specified in this document, there are two distinguishing items that need to be self-generated: the System ID and Router-Fingerprint. In a network device, normally there are some resources which can provide an extremely high probability of uniqueness (some examples listed below). These resources can be used as seeds to derive identifiers.

- MAC address(es)
- Configured IP address(es)
- Hardware IDs (e.g. CPU ID)
- Device serial number(s)
- System clock at a certain specific time
- Arbitrary received packet(s) on an interface(s)

This document recommends the use of an IEEE 802 48-bit MAC address associated with the router as the initial System ID. This document does not specify a specific method to re-generate the System ID when duplication happens.

This document also does not specify a specific method to generate the Router-Fingerprint.

There is an important concern that the seeds listed above (except MAC address) might not be available in some small devices such as home routers. This is because of hardware/software limitations and the lack of sufficient communication packets at the initial stage in home routers when doing ISIS auto-configuration. In this case, this document suggests using the MAC address as System ID and generating a pseudo-random number based on another seed (such as the memory address of a certain variable in the program) as the Router-Fingerprint. The pseudo-random number might not have a very high probability of uniqueness in this solution, but should be sufficient in home networks scenarios.

The considerations surrounding System ID stability described in section Section 3.2 also need to be applied.

3.4.6. Duplication of both System ID and Router-Fingerprint

As described above, the resources for generating System ID/ Fingerprint might be very constrained during the initial stages. Hence, the duplication of both System ID and Router-Fingerprint needs to be considered. In such a case it is possible that a router will receive an LSP with System ID and Router-Fingerprint identical to the local values but the LSP is NOT identical to the locally generated copy i.e. sequence number is newer or sequence number is the same but the LSP has a valid checksum which does not match. The term DD-LSP is used to describe such an LSP.

In a benign case, this will occur if a router restarts and it receives copies of its own LSPs from its previous incarnation. This benign case needs to be distinguished from the pathological case where there are two different routers with the same System ID and the same Router-Fingerprint.

In the benign case, the restarting router will generate a new version of its own LSP with higher sequence number and flood the new LSP version. This will cause other routers in the network to update their LSPDB and synchronization will be achieved.

In the pathological case the generation of a new version of an LSP by one of the "twins" will cause the other twin to generate the same LSP with a higher sequence number - and oscillation will continue without achieving LSPDB synchronization.

Note that comparison of S bit in the Router-Fingerprint TLV cannot be performed as in the benign case it is expected that the S bit will be clear. Also note that the conditions for detecting duplicate System ID will NOT be satisfied because both the System ID and the Router-Fingerprint will be identical.

The following procedure is defined:

```
    DD-state is a boolean which indicates if a
      DD-LSP #0 has been received
    DD-count is the count of the number of occurences
      of reception of a DD-LSP
    DD-timer is a timer associated with reception of
     DD-LSPs. Recommended value is 60 seconds.
    DD-max is the maximum number of DD-LSPs allowed
     to be received in DD-timer interval.
     Recommended value is 3.
```

When a DD-LSP is received:

```
  If DD-state is FALSE:
    DD-state is set to TRUE
    DD-timer is started
    DD-count is initialized to 1.

  If DD-state is TRUE:
    DD-count is incremented
    If DD-count is >= DD-max:
       Local system MUST generate a new System ID
        and Router-Fingerprint and restart the protocol
       DD-state is (re)initialized to FALSE and
        DD-timer cancelled.

  If DD-timer expires:
    DD-state is set to FALSE.
```

Note that to minimze the likelihood of duplication of both System ID and Router-fingerprint reoccuring, routers SHOULD have more entropies available. One simple way to achieve this is to add the LSP sequence number of the next LSP it will send to the Router-Fingerprint.

3.5. Additional IS-IS TLVs Usage Guidelines

This section describes the behavior of selected TLVs when used by a router supporting IS-IS auto-configuration.

3.5.1. Authentication TLV

It is RECOMMENDED that IS-IS routers supporting this specification offer an option to explicitly configure a single password for HMAC-MD5 authentication as specified in[RFC5304].

3.5.2. Metric Used in Reachability TLVs

It is RECOMMENDED that IS-IS auto-configuration routers use a high metric value (e.g. 100000) as default in order to allow manually configured adjacencies to be preferred over auto-configured.

3.5.3. Dynamic Host Name TLV

IS-IS auto-configuration routers MAY advertise their Dynamic Host Name TLV (TLV 137, [RFC5301]). The host name could be provisioned by an IT system, or just use the name of vendor, device type or serial number, etc.

To guarantee the uniqueness of the host name, the System ID SHOULD be appended as a suffix in the names.

4. Security Considerations

In the absence of cryptographic authentication it is possible for an attacker to inject a PDU falsely indicating there is a duplicate system-id. This may trigger automatic restart of the protocol using the duplicate-id resolution procedures defined in this document.

Note that the use of authentication is incompatible with auto-configuration as it requires some manual configuration.

For wired deployment, the wired connection itself could be considered as an implicit authentication in that unwanted routers are usually not able to connect (i.e. there is some kind of physical security in place preventing the connection of rogue devices); for wireless deployment, the authentication could be achieved at the lower wireless link layer.

5. IANA Considerations

This document requires the definition of a new IS-IS TLV to be reflected in the "IS-IS TLV Codepoints" registry:

Type	Description	IIH	LSP	SNP	Purge
TBA	Router-Fingerprint	Y	Y	N	Y

6. Acknowledgements

This document was heavily inspired by [RFC7503].

Martin Winter, Christian Franke and David Lamparter gave essential feedback to improve the technical design based on their implementation experience.

Many useful comments were made by Acee Lindem, Karsten Thomann, Hannes Gredler, Peter Lothberg, Uma Chundury, Qin Wu, Sheng Jiang and Nan Wu, etc.

This document was produced using the xml2rfc tool [RFC7991]. (initially prepared using 2-Word-v2.0.template.dot.)

7. References

7.1. Normative References

[ISO_IEC10589] "Intermediate system to Intermediate system intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473), ISO/IEC 10589:2002, Second Edition.", Nov 2002.

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

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

[RFC5301] McPherson, D. and N. Shen, "Dynamic Hostname Exchange Mechanism for IS-IS", RFC 5301, DOI 10.17487/RFC5301, October 2008, <http://www.rfc-editor.org/info/rfc5301>.

[RFC5304] Li, T. and R. Atkinson, "IS-IS Cryptographic Authentication", RFC 5304, DOI 10.17487/RFC5304, October 2008, <http://www.rfc-editor.org/info/rfc5304>.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <http://www.rfc-editor.org/info/rfc5305>.

[RFC5308] Hopps, C., "Routing IPv6 with IS-IS", RFC 5308, DOI 10.17487/RFC5308, October 2008, <http://www.rfc-editor.org/info/rfc5308>.

7.2. Informative References

[RFC7503] Lindem, A. and J. Arkko, "OSPFv3 Autoconfiguration", RFC 7503, DOI 10.17487/RFC7503, April 2015, <http://www.rfc-editor.org/info/rfc7503>.

[RFC7991] Hoffman, P., "The "xml2rfc" Version 3 Vocabulary", RFC 7991, DOI 10.17487/RFC7991, December 2016, <http://www.rfc-editor.org/info/rfc7991>.

Authors' Addresses

Bing Liu (editor)
Huawei Technologies
Q10, Huawei Campus, No.156 Beiqing Road
Hai-Dian District, Beijing, 100095
P.R. China

Email: leo.liubing@huawei.com

Les Ginsberg
Cisco Systems
821 Alder Drive
Milpitas CA 95035
USA

Email: ginsberg@cisco.com

Bruno Decraene
Orange
France

Email: bruno.decraene@orange.com

Ian Farrer
Deutsche Telekom AG
Bonn
Germany

Email: ian.farrer@telekom.de

Mikael Abrahamsson
T-Systems
Stockholm
Sweden

Email: mikael.abrahamsson@t-systems.se

Networking Working Group N. Shen
Internet-Draft Cisco Systems
Intended status: Standards Track S. Amante
Expires: June 6, 2019 Apple, Inc.
M. Abrahamsson
T-Systems Nordic
December 3, 2018

IS-IS Routing with Reverse Metric
draft-ietf-isis-reverse-metric-17

Abstract

This document describes a mechanism to allow IS-IS routing to quickly and accurately shift traffic away from either a point-to-point or multi-access LAN interface during network maintenance or other operational events. This is accomplished by signaling adjacent IS-IS neighbors with a higher reverse metric, i.e., the metric towards the signaling IS-IS router.

Table of Contents

1. Introduction

The IS-IS [ISO10589] routing protocol has been widely used in Internet Service Provider IP/MPLS networks. Operational experience with the protocol, combined with ever increasing requirements for lossless operations have demonstrated some operational issues. This document describes the issues and a mechanism for mitigating them.

This document defines the IS-IS "Reverse Metric" mechanism that allows an IS-IS node to send a "Reverse Metric" TLV through the IS-IS Hello (IIH) PDU to the neighbor or pseudo-node to adjust the routing metric on the inbound direction.

1.1. Node and Link Isolation

IS-IS routing mechanism has the overload-bit, which can be used by operators to perform disruptive maintenance on the router. But in many operational maintenance cases, it is not necessary to divert all the traffic away from this node. It is necessary to avoid only a single link during the maintenance. More detailed descriptions of the challenges can be found in Appendix A and Appendix B of this document.

1.2. Distributed Forwarding Planes

In a distributed forwarding platform, different forwarding line-cards may have interfaces and IS-IS connections to neighbor routers. If one of the line-card's software resets, it may take some time for the forwarding entries to be fully populated on the line-card, in particular if the router is a PE (Provider Edge) router in ISP's MPLS VPN. An IS-IS adjacency may be established with a neighbor router long before the entire BGP VPN prefixes are downloaded to the forwarding table. It is important to signal to the adjacent IS-IS routers to raise metric values and not to use the corresponding IS-IS adjacency inbound to this router if possible. Temporarily signaling the 'Reverse Metric' over this link to discourage the traffic via the corresponding line-card will help to reduce the traffic loss in the network. In the meantime, the remote PE routers will select a different set of PE routers for the BGP best path calculation or use a different link towards the same PE router on which a line-card is resetting.

1.3. Spine-Leaf Applications

In the IS-IS Spine-Leaf extension [I-D.shen-isis-spine-leaf-ext], the leaf nodes will perform equal-cost or unequal-cost load sharing towards all the spine nodes. In certain operational cases, for instance, when one of the backbone links on a spine node is congested, a spine node can push a higher metric towards the connected leaf nodes to reduce the transit traffic through the corresponding spine node or link.

1.4. LDP IGP Synchronization

In the [RFC5443], a mechanism is described to achieve LDP IGP synchronization by using the maximum link metric value on the interface. But in the case of a new IS-IS node joining the broadcast network (LAN), it is not optimal to change all the nodes on the LAN to the maximum link metric value, as described in [RFC6138]. In this case, the Reverse Metric can be used to discourage both outbound and inbound traffic without affecting the traffic of other IS-IS nodes on the LAN.

1.5. IS-IS Reverse Metric

This document uses the routing protocol itself as the transport mechanism to allow one IS-IS router to advertise a "reverse metric" in an IS-IS Hello (IIH) PDU to an adjacent node on a point-to-point or multi-access LAN link. This would allow the provisioning to be performed only on a single node, setting a "reverse metric" on a link and have traffic bidirectionally shift away from that link gracefully to alternate, viable paths.

This Reverse Metric mechanism is used for both point-to-point and multi-access LAN links. Unlike the point-to-point links, the IS-IS protocol currently does not have a way to influence the traffic towards a particular node on LAN links. This mechanism provides IS-IS routing the capability of altering traffic in both directions on either a point-to-point link or a multi-access link of an IS-IS node.

The metric value in the "reverse metric" TLV and the Traffic Engineering metric in the sub-TLV being advertised is an offset or relative metric to be added to the existing local link and Traffic Engineering metric values of the receiver, the accumulated metric value is bounded as described in Section 2.

1.6. Specification of Requirements

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

2. IS-IS Reverse Metric TLV

The Reverse Metric TLV is a new TLV to be used inside an IS-IS Hello PDU. This TLV is used to support the IS-IS Reverse Metric mechanism that allows a "reverse metric" to be sent to the IS-IS neighbor.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Type     |    Length     |    Flags      |     Metric
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     Metric  (Continue)         | sub-TLV Len   |Optional sub-TLV
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: Reverse Metric TLV

The Value part of the Reverse Metric TLV is composed of a 3 octet field containing an IS-IS Metric Value, a 1 octet field of Flags, and a 1 octet Reverse Metric sub-TLV length field representing the length of a variable number of sub-TLVs. If the "sub-TLV len" is non-zero, then the Value field MUST also contain one or more sub-TLVs.

The Reverse Metric TLV MAY be present in any IS-IS Hello PDU. A sender MUST only transmit a single Reverse Metric TLV in a IS-IS Hello PDU. If a received IS-IS Hello PDU contains more than one Reverse Metric TLV, an implementation MUST ignore all the Reverse Metric TLVs.

```
TYPE: 16
LENGTH: variable (5 - 255 octets)
VALUE:

   Flags (1 octet)
   Metric (3 octets)
   sub-TLV length (1 octet)
   sub-TLV data (0 - 250 octets)

    0 1 2 3 4 5 6 7
   +-+-+-+-+-+-+-+-+
   |  Reserved |U|W|
   +-+-+-+-+-+-+-+-+
```

Figure 2: Flags

The Metric field contains a 24-bit unsigned integer. This value is a metric offset that a neighbor SHOULD add to the existing, configured Default Metric for the IS-IS link [ISO10589]. Refer to "Elements of Procedure", in Section 3 for details on how an IS-IS router should process the Metric field in a Reverse Metric TLV.

The Metric field, in the Reverse Metric TLV, is a "reverse offset metric" that will either be in the range of 0 - 63 when a "narrow" IS-IS metric is used (IS Neighbors TLV, Pseudonode LSP) [RFC1195] or in the range of 0 - (2^24 - 2) when a "wide" Traffic Engineering metric value is used, (Extended IS Reachability TLV) [RFC5305] [RFC5817]. As described below, when the U bit is set, the accumulated value of the wide metric is in the range of 0 - (2^24 - 1), with (2^24 - 1) metric as non-reachable in IS-IS routing. The IS-IS metric value of (2^24 - 2) serves as the link of last resort.

There are currently only two Flag bits defined.

W bit (0x01): The "Whole LAN" bit is only used in the context of multi-access LANs. When a Reverse Metric TLV is transmitted from a node to the Designated Intermediate System (DIS), if the "Whole LAN" bit is set (1), then a DIS SHOULD add the received Metric value in the Reverse Metric TLV to each node's existing Default Metric in the Pseudonode LSP. If the "Whole LAN" bit is not set (0), then a DIS SHOULD add the received Metric value in the Reverse Metric TLV to the existing "default metric" in the Pseudonode LSP for the single node from whom the Reverse Metric TLV was received. Please refer to "Multi-Access LAN Procedures", in Section 3.3, for additional details. The W bit MUST be clear when a Reverse Metric TLV is transmitted in an IIH PDU on a point-to-point link, and MUST be ignored when received on a point-to-point link.

U bit (0x02): The "Unreachable" bit specifies that the metric calculated by addition of the reverse metric to the "default metric" is limited to the maximum value of (2^24-1). This "U" bit applies to both the default metric in the Extended IS Reachability TLV and the Traffic Engineering Default Metric sub-TLV of the link. This is only relevant to the IS-IS "wide" metric mode.

The Reserved bits of Flags field MUST be set to zero and MUST be ignored when received.

The Reverse Metric TLV MAY include sub-TLVs when an IS-IS router wishes to signal additional information to its neighbor. In this document, the Reverse Metric Traffic Engineering Metric sub-TLV, with Type 18, is defined. This Traffic Engineering Metric contains a 24-bit unsigned integer. This sub-TLV is optional, if it appears more than once, then the entire Reverse Metric TLV MUST be ignored. Upon receiving this Traffic Engineering METRIC sub-TLV in a Reverse Metric TLV, a node SHOULD add the received Traffic Engineering Metric offset value to its existing, configured Traffic Engineering Default Metric within its Extended IS Reachability TLV. The use of other sub-TLVs is outside the scope of this document. The "sub-TLV Len" value MUST be set to zero when an IS-IS router does not have Traffic Engineering sub-TLVs that it wishes to send to its IS-IS neighbor.

3. Elements of Procedure

3.1. Processing Changes to Default Metric

It is important to use the same IS-IS metric type on both ends of the link and in the entire IS-IS area or level. On the receiving side of the 'reverse-metric' TLV, the accumulated value of configured metric and the reverse-metric needs to be limited to 63 in "narrow" metric mode and to (2^24 - 2) in "wide" metric mode. This applies to both the Default Metric of Extended IS Reachability TLV and the Traffic Engineering Default Metric sub-TLV in LSP or Pseudonode LSP for the "wide" metric mode case. If the "U" bit is present in the flags, the accumulated metric value is to be limited to (2^24 - 1) for both the normal link metric and Traffic Engineering metric in IS-IS "wide" metric mode.

If an IS-IS router is configured to originate a Traffic Engineering Default Metric sub-TLV for a link, but receives a Reverse Metric TLV from its neighbor that does not contain a Traffic Engineering Default Metric sub-TLV, then the IS-IS router MUST NOT change the value of its Traffic Engineering Default Metric sub-TLV for that link.

3.2. Multi-Topology IS-IS Support on Point-to-point links

The Reverse Metric TLV is applicable to Multi-Topology IS-IS (M-ISIS) [RFC5120]. On point-to-point links, if an IS-IS router is configured for M-ISIS, it MUST send only a single Reverse Metric TLV in IIH PDUs toward its neighbor(s) on the designated link. When an M-ISIS router receives a Reverse Metric TLV, it MUST add the received Metric value to its Default Metric of the link in all Extended IS Reachability TLVs for all topologies. If an M-ISIS router receives a Reverse Metric TLV with a Traffic Engineering Default Metric sub-TLV, then the M-ISIS router MUST add the received Traffic Engineering Default Metric value to each of its Default Metric sub-TLVs in all of its MT Intermediate Systems TLVs. If an M-ISIS router is configured to advertise Traffic Engineering Default Metric sub-TLVs for one or more topologies, but does not receive a Traffic Engineering Default Metric sub-TLV in a Reverse Metric TLV, then the M-ISIS router MUST NOT change the value in each of the Traffic Engineering Default Metric sub-TLVs for all topologies.

3.3. Multi-Access LAN Procedures

On a Multi-Access LAN, only the DIS SHOULD act upon information contained in a received Reverse Metric TLV. All non-DIS nodes MUST silently ignore a received Reverse Metric TLV. The decision process of the routers on the LAN MUST follow the procedure in section

7.2.8.2 of [ISO10589], and use the "Two-way connectivity check" during the topology and route calculation.

The Reverse Metric Traffic Engineering sub-TLV also applies to the DIS. If a DIS is configured to apply Traffic Engineering over a link and it receives Traffic Engineering Metric sub-TLV in a Reverse Metric TLV, it should update the Traffic Engineering Default Metric sub-TLV value of the corresponding Extended IS Reachability TLV or insert a new one if not present.

In the case of multi-access LANs, the "W" Flags bit is used to signal from a non-DIS to the DIS whether to change the metric and, optionally, Traffic Engineering parameters for all nodes in the Pseudonode LSP or solely the node on the LAN originating the Reverse Metric TLV.

A non-DIS node, e.g., Router B, attached to a multi-access LAN will send the DIS a Reverse Metric TLV with the W bit clear when Router B wishes the DIS to add the Metric value to the Default Metric contained in the Pseudonode LSP specific to just Router B. Other non-DIS nodes, e.g., Routers C and D, may simultaneously send a Reverse Metric TLV with the W bit clear to request the DIS to add their own Metric value to their Default Metric contained in the Pseudonode LSP.

As long as at least one IS-IS node on the LAN sending the signal to DIS with the W bit set, the DIS would add the metric value in the Reverse Metric TLV to all neighbor adjacencies in the Pseudonode LSP, regardless if some of the nodes on the LAN advertise the Reverse Metric TLV without the W bit set. The DIS MUST use the reverse metric of the highest source MAC address Non-DIS advertising the Reverse Metric TLV with the W bit set.

Local provisioning on the DIS to adjust the Default Metric(s) is another way to insert Reverse Metric in the Pseudonode LSP towards an IS-IS node on a LAN. In the case where Reverse Metric TLV is also used in the IS-IS Hello PDU of the node, the local provisioning MUST take precedence over received Reverse Metric TLVs. For instance, local policy on the DIS may be provisioned to ignore the W bit signaling on a LAN.

Multi-Topology IS-IS [RFC5120] specifies there is no change to construction of the Pseudonode LSP, regardless of the Multi-Topology capabilities of a multi-access LAN. If any MT capable node on the LAN advertises the Reverse Metric TLV to the DIS, the DIS should update, as appropriate, the Default Metric contained in the Pseudonode LSP. If the DIS updates the Default Metric in and floods a new Pseudonode LSP, those default metric values will be applied to all topologies during Multi-Topology SPF calculations.

3.4. LDP/IGP Synchronization on LANs

As described in [RFC6138] when a new IS-IS node joins a broadcast network, it is unnecessary and sometimes even harmful for all IS-IS nodes on the LAN to advertise maximum link metric. [RFC6138] proposes a solution to have the new node not advertise its adjacency towards the pseudo-node when it is not in a "cut-edge" position.

With the introduction of Reverse Metric in this document, a simpler alternative solution to the above mentioned problem can be used. The Reverse Metric allows the new node on the LAN to advertise its inbound metric value to be the maximum and this puts the link of this new node in the last resort position without impacting the other IS-IS nodes on the same LAN.

Specifically, when IS-IS adjacencies are being established by the new node on the LAN, besides setting the maximum link metric value (2^24 - 2) on the interface of the LAN for LDP IGP synchronization as described in [RFC5443], it SHOULD advertise the maximum metric offset value in the Reverse Metric TLV in its IIH PDU sent on the LAN. It SHOULD continue this advertisement until it completes all the LDP label binding exchanges with all the neighbors over this LAN, either by receiving the LDP End-of-LIB [RFC5919] for all the sessions or by exceeding the provisioned timeout value for the node LDP/IGP synchronization.

3.5. Operational Guidelines

For the use case in Section 1.1, a router SHOULD limit the period of advertising a Reverse Metric TLV towards a neighbor only for the duration of network maintenance window.

The use of Reverse Metric does not alter IS-IS metric parameters stored in a router’s persistent provisioning database.

If routers that receive a Reverse Metric TLV sends a syslog message or SNMP trap, this will assist in rapidly identifying the node in the network that is advertising an IS-IS metric or Traffic Engineering parameters different from that which is configured locally on the device.

When the link Traffic Engineering metric is raised to (2^24 - 1) [RFC5817], either due to the reverse-metric mechanism or by explicit user configuration, this SHOULD immediately trigger the CSPF (Constrained Shortest Path First) re-calculation to move the Traffic

Engineering traffic away from that link. It is RECOMMENDED also that the CSPF does the immediate CSPF re-calculation when the Traffic Engineering metric is raised to (2^24 - 2) to be the last resort link.

It is advisable that implementations provide a configuration capability to disable any IS-IS metric changes by Reverse Metric mechanism through neighbor's Hello PDUs.

If an implementation enables this mechanism by default, it is RECOMMENDED that it be disabled by the operators when not explicitly using it.

4. Security Considerations

Security concerns for IS-IS are addressed in [ISO10589], [RFC5304], [RFC5310], and with various deployment and operational security considerations in [RFC7645]. The enhancement in this document makes it possible for one IS-IS router to manipulate the IS-IS Default Metric and, optionally, Traffic Engineering parameters of adjacent IS-IS neighbors on point-to-point or LAN interfaces. Although IS-IS routers within a single Autonomous System nearly always are under the control of a single administrative authority, it is highly recommended that operators configure authentication of IS-IS PDUs to mitigate use of the Reverse Metric TLV as a potential attack vector.

5. IANA Considerations

IANA has allocated IS-IS TLV Codepoints of 16 for the Reverse Metric TLV. This new TLV has the following attributes: IIH = y, LSP = n, SNP = n, Purge = n.

This document also introduces a new registry for sub-TLVs of the Reverse Metric TLV. The registration policy is Expert Review as defined in [RFC8126]. This registry is part of the "IS-IS TLV Codepoints" registry. The name of the registry is "Sub-TLVs for Reverse Metric TLV". The defined values are:

```
0:        Reserved
1-17:     Unassigned
18:       Traffic Engineering Metric sub-TLV, as specified in this
          document (Section 2)
19-255:   Unassigned
```

6. Acknowledgments

The authors would like to thank Mike Shand, Dave Katz, Guan Deng, Ilya Varlashkin, Jay Chen, Les Ginsberg, Peter Ashwood-Smith, Uma Chunduri, Alexander Okonnikov, Jonathan Harrison, Dave Ward, Himanshu Shah, Wes George, Danny McPherson, Ed Crabbe, Russ White, Robert Raszuk, Tom Petch, Stewart Bryant and Acee Lindem for their comments and contributions.

This document was produced using Marshall Rose's xml2rfc tool.

7. References

7.1. Normative References

[ISO10589]
ISO, "Intermediate system to Intermediate system routeing information exchange protocol for use in conjunction with the Protocol for providing the Connectionless-mode Network Service (ISO 8473)", ISO/IEC 10589:2002.

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

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

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

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <https://www.rfc-editor.org/info/rfc5305>.

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

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

7.2. Informative References

[I-D.shen-isis-spine-leaf-ext] Shen, N., Ginsberg, L., and S. Thyamagundalu, "IS-IS Routing for Spine-Leaf Topology", draft-shen-isis-spine-leaf-ext-07 (work in progress), October 2018.

[RFC5304] Li, T. and R. Atkinson, "IS-IS Cryptographic Authentication", RFC 5304, DOI 10.17487/RFC5304, October 2008, <https://www.rfc-editor.org/info/rfc5304>.

[RFC5310] Bhatia, M., Manral, V., Li, T., Atkinson, R., White, R., and M. Fanto, "IS-IS Generic Cryptographic Authentication", RFC 5310, DOI 10.17487/RFC5310, February 2009, <https://www.rfc-editor.org/info/rfc5310>.

[RFC5817] Ali, Z., Vasseur, JP., Zamfir, A., and J. Newton, "Graceful Shutdown in MPLS and Generalized MPLS Traffic Engineering Networks", RFC 5817, DOI 10.17487/RFC5817, April 2010, <https://www.rfc-editor.org/info/rfc5817>.

[RFC5919] Asati, R., Mohapatra, P., Chen, E., and B. Thomas, "Signaling LDP Label Advertisement Completion", RFC 5919, DOI 10.17487/RFC5919, August 2010, <https://www.rfc-editor.org/info/rfc5919>.

[RFC6138] Kini, S., Ed. and W. Lu, Ed., "LDP IGP Synchronization for Broadcast Networks", RFC 6138, DOI 10.17487/RFC6138, February 2011, <https://www.rfc-editor.org/info/rfc6138>.

[RFC7645] Chunduri, U., Tian, A., and W. Lu, "The Keying and Authentication for Routing Protocol (KARP) IS-IS Security Analysis", RFC 7645, DOI 10.17487/RFC7645, September 2015, <https://www.rfc-editor.org/info/rfc7645>.

Appendix A. Node Isolation Challenges

On rare occasions, it is necessary for an operator to perform disruptive network maintenance on an entire IS-IS router node, i.e., major software upgrades, power/cooling augments, etc. In these cases, an operator will set the IS-IS Overload Bit (OL-bit) within the Link State Protocol Data Units (LSPs) of the IS-IS router about to undergo maintenance. The IS-IS router immediately floods its updated LSPs to all IS-IS routers in the IS-IS domain. Upon receipt of the updated LSPs, all IS-IS routers recalculate their Shortest Path First (SPF) tree excluding IS-IS routers whose LSPs have the OL-bit set. This effectively removes the IS-IS router about to undergo maintenance from the topology, thus preventing it from receiving any transit traffic during the maintenance period.

After the maintenance activity has completed, the operator resets the IS-IS Overload Bit within the LSPs of the original IS-IS router causing it to flood updated IS-IS LSPs throughout the IS-IS domain. All IS-IS routers recalculate their SPF tree and now include the original IS-IS router in their topology calculations, allowing it to be used for transit traffic again.

Isolating an entire IS-IS router from the topology can be especially disruptive due to the displacement of a large volume of traffic through an entire IS-IS router to other, sub-optimal paths, (e.g., those with significantly larger delay). Thus, in the majority of network maintenance scenarios, where only a single link or LAN needs to be augmented to increase its physical capacity or is experiencing an intermittent failure, it is much more common and desirable to gracefully remove just the targeted link or LAN from service, temporarily, so that the least amount of user-data traffic is affected during the link-specific network maintenance.

Appendix B. Link Isolation Challenges

Before network maintenance events are performed on individual physical links or LANs, operators substantially increase the IS-IS metric simultaneously on both devices attached to the same link or LAN. In doing so, the devices generate new Link State Protocol Data Units (LSPs) that are flooded throughout the network and cause all routers to gradually shift traffic onto alternate paths with very little or no disruption to in-flight communications by applications or end-users. When performed successfully, this allows the operator to confidently perform disruptive augmentation, fault diagnosis or repairs on a link without disturbing ongoing communications in the network.

There are a number of challenges with the above solution. First, it is quite common to have routers with several hundred interfaces and individual interfaces that are from several hundred Gigabits/second to Terabits/second of traffic. Thus, it is imperative that operators accurately identify the same point-to-point link on two, separate devices in order to increase (and, afterward, decrease) the IS-IS metric appropriately. Second, the aforementioned solution is very time consuming and even more error-prone to perform when it’s necessary to temporarily remove a multi-access LAN from the network topology. Specifically, the operator needs to configure ALL devices that have interfaces attached to the multi-access LAN with an appropriately high IS-IS metric, (and then decrease the IS-IS metric to its original value afterward). Finally, with respect to multi-access LANs, there is currently no method to bidirectionally isolate only a single node’s interface on the LAN when performing more fine-grained diagnosis and repairs to the multi-access LAN.

In theory, use of a Network Management System (NMS) could improve the accuracy of identifying the appropriate subset of routers attached to either a point-to-point link or a multi-access LAN as well as signaling from the NMS to those devices, using a network management protocol to adjust the IS-IS metrics on the pertinent set of interfaces. The reality is that NMSs are, to a very large extent, not used within Service Provider’s networks for a variety of reasons. In particular, NMSs do not interoperate very well across different vendors or even separate platform families within the same vendor.

Appendix C. Contributors’ Addresses

Tony Li

Email: tony.li@tony.li

Authors’ Addresses

Naiming Shen
Cisco Systems
560 McCarthy Blvd.
Milpitas, CA 95035
USA

Email: naiming@cisco.com

Shane Amante
Apple, Inc.
1 Infinite Loop
Cupertino, CA 95014
USA

Email: samante@apple.com

Mikael Abrahamsson
T-Systems Nordic
Kistagangen 26
Stockholm
SE

Email: Mikael.Abrahamsson@t-systems.se

Cisco Systems, Inc.
A. Bashandy
Arrcus
H. Gredler
RtBrick Inc.
B. Decraene
Orange
May 19, 2019

IS-IS Extensions for Segment Routing
draft-ietf-isis-segment-routing-extensions-25

Abstract

Segment Routing (SR) allows for a flexible definition of end-to-end paths within IGP topologies by encoding paths as sequences of topological sub-paths, called "segments". These segments are advertised by the link-state routing protocols (IS-IS and OSPF).

This draft describes the necessary IS-IS extensions that need to be introduced for Segment Routing operating on an MPLS data-plane.

Requirements Language

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

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Segment Routing (SR) allows for a flexible definition of end-to-end paths within IGP topologies by encoding paths as sequences of topological sub-paths, called "segments". These segments are advertised by the link-state routing protocols (IS-IS and OSPF). Prefix segments represent an ECMP-aware shortest-path to a prefix (or a node), as per the state of the IGP topology. Adjacency segments represent a hop over a specific adjacency between two nodes in the IGP. A prefix segment is typically a multi-hop path while an adjacency segment, in most of the cases, is a one-hop path. SR's control-plane can be applied to both IPv6 and MPLS data-planes, and does not require any additional signaling (other than the regular IGP). For example, when used in MPLS networks, SR paths do not require any LDP or RSVP-TE signaling. Still, SR can interoperate in the presence of LSPs established with RSVP or LDP.

There are additional segment types, e.g., Binding SID defined in [RFC8402]. This document also defines an advertisement for one type of Binding SID: the Mirror Context segment.

This draft describes the necessary IS-IS extensions that need to be introduced for Segment Routing operating on an MPLS data-plane.

The Segment Routing architecture is described in [RFC8402].

Segment Routing use cases are described in [RFC7855].

2. Segment Routing Identifiers

The Segment Routing architecture [RFC8402] defines different types of Segment Identifiers (SID). This document defines the IS-IS encodings for the IGP-Prefix Segment, the IGP-Adjacency Segment, the IGP-LAN-Adjacency Segment and the Binding Segment.

2.1. Prefix Segment Identifier (Prefix-SID Sub-TLV)

A new IS-IS sub-TLV is defined: the Prefix Segment Identifier sub-TLV (Prefix-SID sub-TLV).

The Prefix-SID sub-TLV carries the Segment Routing IGP-Prefix-SID as defined in [RFC8402]. The 'Prefix SID' MUST be unique within a given IGP domain (when the L-flag is not set).

A Prefix-SID sub-TLV is associated to a prefix advertised by a node and MAY be present in any of the following TLVs:

TLV-135 (Extended IPv4 reachability) defined in [RFC5305].

TLV-235 (Multitopology IPv4 Reachability) defined in [RFC5120].

TLV-236 (IPv6 IP Reachability) defined in [RFC5308].

TLV-237 (Multitopology IPv6 IP Reachability) defined in [RFC5120].

Binding-TLV and Multi-Topology Binding-TLV defined in Section 2.4 and Section 2.5 respectively.

The Prefix-SID sub-TLV has the following format:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Type        |     Length    |     Flags     |   Algorithm   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     SID/Index/Label (variable)                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Type: 3

Length: 5 or 6 depending on the size of the SID (described below)

Flags: 1 octet field of following flags:

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|R|N|P|E|V|L|   |
+-+-+-+-+-+-+-+-+
```

where:

R-Flag: Re-advertisement flag. If set, then the prefix to which this Prefix-SID is attached, has been propagated by the router either from another level (i.e., from level-1 to level-2 or the opposite) or from redistribution (e.g.: from another protocol).

N-Flag: Node-SID flag. If set, then the Prefix-SID refers to the router identified by the prefix. Typically, the N-Flag is set on Prefix-SIDs attached to a router loopback address. The N-Flag is set when the Prefix-SID is a Node-SID as described in [RFC8402].

P-Flag: no-PHP flag. If set, then the penultimate hop MUST NOT pop the Prefix-SID before delivering the packet to the node that advertised the Prefix-SID.

E-Flag: Explicit-Null Flag. If set, any upstream neighbor of the Prefix-SID originator MUST replace the Prefix-SID with a Prefix-SID having an Explicit-NULL value (0 for IPv4 and 2 for IPv6) before forwarding the packet.

V-Flag: Value flag. If set, then the Prefix-SID carries a value (instead of an index). By default the flag is UNSET.

L-Flag: Local Flag. If set, then the value/index carried by the Prefix-SID has local significance. By default the flag is UNSET.

Other bits: MUST be zero when originated and ignored when received.

Algorithm: the router may use various algorithms when calculating reachability to other nodes or to prefixes attached to these nodes. Algorithm identifiers are defined in Section 3.2. Examples of these algorithms are metric based Shortest Path First (SPF), various sorts of Constrained SPF, etc. The algorithm field of the Prefix-SID contains the identifier of the algorithm the router uses to compute the reachability of the prefix to which the Prefix-SID is associated.

At origination, the Prefix-SID algorithm field MUST be set to 0 or to any value advertised in the SR-Algorithm sub-TLV (Section 3.2).

A router receiving a Prefix-SID from a remote node and with an algorithm value that such remote node has not advertised in the SR-Algorithm sub-TLV (Section 3.2) MUST ignore the Prefix-SID sub-TLV.

SID/Index/Label as defined in Section 2.1.1.1.

When the Prefix SID is an index (the V-flag is not set) the value is used to determine the actual label value inside the set of all advertised label ranges of a given router. This allows a receiving router to construct forwarding state to a particular destination router.

In many use-cases a 'stable transport' address is overloaded as an identifier of a given node. Because Prefixes may be re-advertised into other levels there may be some ambiguity (e.g. Originating router vs. L1L2 router) for which node a particular IP prefix serves as identifier. The Prefix-SID sub-TLV contains the necessary flags to disambiguate Prefix to node mappings. Furthermore if a given node has several 'stable transport' addresses there are flags to differentiate those among other Prefixes advertised from a given node.

2.1.1. Flags

2.1.1.1. V and L Flags

The V-flag indicates whether the SID/Index/Label field is a value or an index.

The L-Flag indicates whether the value/index in the SID/Index/Label field has local or global significance.

The following settings for V and L flags are valid:

V-flag is set to 0 and L-flag is set to 0: The SID/Index/Label field is a 4 octet index defining the offset in the SID/Label space advertised by this router using the encodings defined in Section 3.1.

V-flag is set to 1 and L-flag is set to 1: The SID/Index/Label field is a 3 octet local label where the 20 rightmost bits are used for encoding the label value.

All other combinations of V-flag and L-flag are invalid and any SID advertisement received with an invalid setting for V and L flags MUST be ignored.

2.1.1.2. R and N Flags

The R-Flag MUST be set for prefixes that are not local to the router and either:

advertised because of propagation (Level-1 into Level-2);

advertised because of leaking (Level-2 into Level-1);

advertised because of redistribution (e.g.: from another protocol).

In the case where a Level-1-2 router has local interface addresses configured in one level, it may also propagate these addresses into the other level. In such case, the Level-1-2 router MUST NOT set the R bit.

The N-Flag is used in order to define a Node-SID. A router MAY set the N-Flag only if all of the following conditions are met:

The prefix to which the Prefix-SID is attached is local to the router (i.e., the prefix is configured on one of the local interfaces, e.g., a 'stable transport' loopback).

The prefix to which the Prefix-SID is attached has a Prefix length of either /32 (IPv4) or /128 (IPv6).

The router MUST ignore the N-Flag on a received Prefix-SID if the prefix has a Prefix length different than /32 (IPv4) or /128 (IPv6).

The Prefix Attributes Flags sub-TLV [RFC7794] also defines the N and R flags and with the same semantics of the equivalent flags defined in this document. Whenever the Prefix Attributes Flags sub-TLV is present for a given prefix the values of the N and R flags advertised in that sub-TLV MUST be used and the values in a corresponding Prefix SID sub-TLV (if present) MUST be ignored.

2.1.1.3. E and P Flags

The following behavior is associated with the settings of the E and P flags:

o If the P-flag is not set then any upstream neighbor of the Prefix-SID originator MUST pop the Prefix-SID. This is equivalent to the penultimate hop popping mechanism used in the MPLS dataplane which improves performance of the ultimate hop. MPLS EXP bits of the Prefix-SID are not preserved to the ultimate hop (the Prefix-SID being removed). If the P-flag is unset the received E-flag is ignored.

o If the P-flag is set then:

   * If the E-flag is not set then any upstream neighbor of the Prefix-SID originator MUST keep the Prefix-SID on top of the stack. This is useful when, e.g., the originator of the

Prefix-SID must stitch the incoming packet into a continuing MPLS LSP to the final destination. This could occur at an inter-area border router (prefix propagation from one area to another) or at an inter-domain border router (prefix propagation from one domain to another).

* If the E-flag is set then any upstream neighbor of the Prefix-SID originator MUST replace the PrefixSID with a Prefix-SID having an Explicit-NULL value. This is useful, e.g., when the originator of the Prefix-SID is the final destination for the related prefix and the originator wishes to receive the packet with the original EXP bits.

When propagating (either from Level-1 to Level-2 or vice versa) a reachability advertisement originated by another IS-IS speaker, the router MUST set the P-flag and MUST clear the E-flag of the related Prefix-SIDs.

2.1.2. Prefix-SID Propagation

The Prefix-SID sub-TLV MUST be included when the associated Prefix Reachability TLV is propagated across level boundaries.

The level-1-2 router that propagates the Prefix-SID sub-TLV between levels maintains the content (flags and SID) except as noted in Section 2.1.1.2 and Section 2.1.1.3.

2.2. Adjacency Segment Identifier

A new IS-IS sub-TLV is defined: the Adjacency Segment Identifier sub-TLV (Adj-SID sub-TLV).

The Adj-SID sub-TLV is an optional sub-TLV carrying the Segment Routing IGP-Adjacency-SID as defined in [RFC8402] with flags and fields that may be used, in future extensions of Segment Routing, for carrying other types of SIDs.

IS-IS adjacencies are advertised using one of the IS-Neighbor TLVs below:

TLV-22 (Extended IS reachability)[RFC5305]

TLV-222 (Multitopology IS)[RFC5120]

TLV-23 (IS Neighbor Attribute)[RFC5311]

TLV-223 (Multitopology IS Neighbor Attribute)[RFC5311]

TLV-141 (inter-AS reachability information)[RFC5316]

Multiple Adj-SID sub-TLVs MAY be associated with a single IS-neighbor.

2.2.1. Adjacency Segment Identifier (Adj-SID) Sub-TLV

The following format is defined for the Adj-SID sub-TLV:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Type        |     Length    |     Flags     |     Weight    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    SID/Label/Index (variable)                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Type: 31

Length: 5 or 6 depending on size of the SID

Flags: 1 octet field of following flags:

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|F|B|V|L|S|P|   |
+-+-+-+-+-+-+-+-+
```

where:

F-Flag: Address-Family flag. If unset, then the Adj-SID is used when forwarding IPv4 encapsulated traffic to the neighbor. If set then the Adj-SID is used when forwarding IPv6 encapsulated traffic to the neighbor.

B-Flag: Backup flag. If set, the Adj-SID is eligible for protection (e.g.: using IPFRR or MPLS-FRR) as described in [RFC8402].

V-Flag: Value flag. If set, then the Adj-SID carries a value. By default the flag is SET.

L-Flag: Local Flag. If set, then the value/index carried by the Adj-SID has local significance. By default the flag is SET.

S-Flag. Set flag. When set, the S-Flag indicates that the Adj-SID refers to a set of adjacencies (and therefore MAY be assigned to other adjacencies as well).

P-Flag. Persistent flag. When set, the P-Flag indicates that the Adj-SID is persistently allocated, i.e., the Adj-SID value remains consistent across router restart and/or interface flap.

Other bits: MUST be zero when originated and ignored when received.

Weight: 1 octet. The value represents the weight of the Adj-SID for the purpose of load balancing. The use of the weight is defined in [RFC8402].

SID/Index/Label as defined in Section 2.1.1.1.

An SR capable router MAY allocate an Adj-SID for each of its adjacencies

An SR capable router MAY allocate more than one Adj-SID to an adjacency.

An SR capable router MAY allocate the same Adj-SID to different adjacencies.

When the P-flag is not set, the Adj-SID MAY be persistent. When the P-flag is set, the Adj-SID MUST be persistent.

Examples of use of the Adj-SID sub-TLV are described in [RFC8402].

The F-flag is used in order for the router to advertise the outgoing encapsulation of the adjacency the Adj-SID is attached to.

2.2.2. Adjacency Segment Identifiers in LANs

In LAN subnetworks, the Designated Intermediate System (DIS) is elected and originates the Pseudonode-LSP (PN-LSP) including all neighbors of the DIS.

When Segment Routing is used, each router in the LAN MAY advertise the Adj-SID of each of its neighbors. Since, on LANs, each router only advertises one adjacency to the DIS (and doesn’t advertise any other adjacency), each router advertises the set of Adj-SIDs (for each of its neighbors) inside a newly defined sub-TLV part of the TLV advertising the adjacency to the DIS (e.g.: TLV-22).

The following new sub-TLV is defined: LAN-Adj-SID containing the set of Adj-SIDs the router assigned to each of its LAN neighbors.

The format of the LAN-Adj-SID sub-TLV is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Type        |     Length    |     Flags     |    Weight     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Neighbor System-ID (ID length octets)        |
+                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   SID/Label/Index (variable)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Type: 32

Length: variable.

Flags: 1 octet field of following flags:

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|F|B|V|L|S|P|   |
+-+-+-+-+-+-+-+-+
```

where F, B, V, L, S and P flags are defined in Section 2.2.1. Other bits: MUST be zero when originated and ignored when received.

Weight: 1 octet. The value represents the weight of the Adj-SID for the purpose of load balancing. The use of the weight is defined in [RFC8402].

Neighbor System-ID: IS-IS System-ID of length "ID Length" as defined in [ISO10589].

SID/Index/Label as defined in Section 2.1.1.1.

Multiple LAN-Adj-SID sub-TLVs MAY be encoded.

Note that this sub-TLV MUST NOT appear in TLV 141.

In case one TLV-22/23/222/223 (reporting the adjacency to the DIS) can't contain the whole set of LAN-Adj-SID sub-TLVs, multiple advertisements of the adjacency to the DIS MUST be used and all advertisements MUST have the same metric.

Each router within the level, by receiving the DIS PN LSP as well as the non-PN LSP of each router in the LAN, is capable of reconstructing the LAN topology as well as the set of Adj-SIDs each router uses for each of its neighbors.

2.3. SID/Label Sub-TLV

The SID/Label sub-TLV may be present in the following TLVs/sub-TLVs defined in this document:

SR-Capabilities Sub-TLV (Section 3.1)

SR Local Block Sub-TLV (Section 3.3)

SID/Label Binding TLV (Section 2.4)

Multi-Topology SID/Label Binding TLV (Section 2.5)

Note that the code point used in all of the above cases is the SID/ Label Sub-TLV code point specified in the new "sub-TLVs for TLV 149 and 150" registry created by this document.

The SID/Label sub-TLV contains a SID or a MPLS Label. The SID/Label sub-TLV has the following format:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Type        |     Length    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          SID/Label (variable)                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Type: 1

Length: 3 or 4

SID/Label: if length is set to 3 then the 20 rightmost bits represent a MPLS label. If length is set to 4 then the value is a 32 bit index

2.4. SID/Label Binding TLV

The SID/Label Binding TLV MAY be originated by any router in an IS-IS domain. There are multiple uses of the SID/Label Binding TLV.

The SID/Label Binding TLV may be used to advertise prefixes to SID/Label mappings. This functionality is called the Segment Routing Mapping Server (SRMS). The behavior of the SRMS is defined in [I-D.ietf-spring-segment-routing-ldp-interop].

The SID/Label Binding TLV may also be used to advertise a Mirror SID to advertise the ability to process traffic originally destined to another IGP node. This behavior is defined in [RFC8402].

The SID/Label Binding TLV has the following format:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Type     |    Length     |     Flags     |    RESERVED   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Range              | Prefix Length |     Prefix    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//               Prefix (continued, variable)                  //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Sub-TLVs (variable)                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: SID/Label Binding TLV format

- o Type: 149
- o Length: variable.
- o 1 octet of flags
- o 1 octet of RESERVED (SHOULD be transmitted as 0 and MUST be ignored on receipt)
- o 2 octets of Range
- o 1 octet of Prefix Length
- o 0-16 octets of Prefix

o sub-TLVs, where each sub-TLV consists of a sequence of:

   * 1 octet of sub-TLV type

   * 1 octet of length of the value field of the sub-TLV

   * 0-243 octets of value

2.4.1. Flags

Flags: 1 octet field of following flags:

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|F|M|S|D|A|     |
+-+-+-+-+-+-+-+-+
```

where:

F-Flag: Address Family flag. If unset, then the Prefix carries an IPv4 Prefix. If set then the Prefix carries an IPv6 Prefix.

M-Flag: Mirror Context flag. Set if the advertised SID corresponds to a mirrored context. The use of a mirrored context is described in [RFC8402].

S-Flag: If set, the SID/Label Binding TLV SHOULD be flooded across the entire routing domain. If the S flag is not set, the SID/ Label Binding TLV MUST NOT be leaked between levels. This bit MUST NOT be altered during the TLV leaking.

D-Flag: when the SID/Label Binding TLV is leaked from level-2 to level-1, the D-Flag MUST be set. Otherwise, this flag MUST be clear. SID/Label Binding TLVs with the D-Flag set MUST NOT be leaked from level-1 to level-2. This is to prevent TLV looping across levels.

A-Flag: Attached flag. The originator of the SID/Label Binding TLV MAY set the A bit in order to signal that the prefixes and SIDs advertised in the SID/Label Binding TLV are directly connected to their originators. The mechanisms through which the originator of the SID/Label Binding TLV can figure out if a prefix is attached or not are outside the scope of this document (e.g.: through explicit configuration). If the Binding TLV is leaked to other areas/levels the A-flag MUST be cleared.

An implementation may decide not to honor the S-flag in order not to leak Binding TLV’s between levels (for policy reasons).

Other bits: MUST be zero when originated and ignored when received.

2.4.2. Range

The 'Range' field provides the ability to specify a range of addresses and their associated Prefix SIDs. This advertisement supports the SRMS functionality. It is essentially a compression scheme to distribute a continuous Prefix and their continuous, corresponding SID/Label Block. If a single SID is advertised then the range field MUST be set to one. For range advertisements > 1, the range field MUST be set to the number of addresses that need to be mapped into a Prefix-SID. In either case the prefix is the first address to which a SID is to be assigned.

2.4.3. Prefix Length, Prefix

The 'Prefix' represents the Forwarding equivalence class at the tail-end of the advertised path. The 'Prefix' does not need to correspond to a routable prefix of the originating node.

The 'Prefix Length' field contains the length of the prefix in bits. Only the most significant octets of the Prefix are encoded (i.e., 1 octet for prefix length 1 up to 8, 2 octets for prefix length 9 to 16, 3 octets for prefix length 17 up to 24 and 4 octets for prefix length 25 up to 32, ..., 16 octets for prefix length 113 up to 128).

2.4.4. Mapping Server Prefix-SID

The Prefix-SID sub-TLV is defined in Section 2.1 and contains the SID/index/label value associated with the prefix and range. The Prefix-SID Sub-TLV MUST be present in the SID/Label Binding TLV when the M-flag is clear. The Prefix-SID Sub-TLV MUST NOT be present when the M-flag is set.

2.4.4.1. Prefix-SID Flags

The Prefix-SID flags are defined in Section 2.1. The Mapping Server MAY advertise a mapping with the N flag set when the prefix being mapped is known in the link-state topology with a mask length of 32 (IPv4) or 128 (IPv6) and when the prefix represents a node. The mechanisms through which the operator defines that a prefix represents a node are outside the scope of this document (typically it will be through configuration).

The other flags defined in Section 2.1 are not used by the Mapping Server and MUST be ignored at reception.

2.4.4.2. PHP Behavior when using Mapping Server Advertisements

As the mapping server does not specify the originator of a prefix advertisement it is not possible to determine PHP behavior solely based on the Mapping Server Advertisement. However, if additional information is available PHP behavior may safely be done. The required information consists of:

o A prefix reachability advertisement for the prefix has been received which includes the Prefix Attribute Flags sub-TLV [RFC7794].

o X and R flags are both set to 0 in the Prefix Attribute Flags sub-TLV.

In the absence of an Prefix Attribute Flags sub-TLV [RFC7794] the A flag in the binding TLV indicates that the originator of a prefix reachability advertisement is directly connected to the prefix and thus PHP MUST be done by the neighbors of the router originating the prefix reachability advertisement. Note that A-flag is only valid in the original area in which the Binding TLV is advertised.

2.4.4.3. Prefix-SID Algorithm

The algorithm field contains the identifier of the algorithm associated with the SIDs for the prefix(es) in the range. Use of the algorithm field is described in Section 2.1.

2.4.5. SID/Label Sub-TLV

The SID/Label sub-TLV (Type: 1) contains the SID/Label value as defined in Section 2.3. It MUST be present in the SID/Label Binding TLV when the M-flag is set in the Flags field of the parent TLV.

2.4.6. Example Encodings

Example 1: if the following IPv4 router addresses (loopback addresses) need to be mapped into the corresponding Prefix SID indexes.

```
Router-A: 192.0.2.1/32, Prefix-SID: Index 1
Router-B: 192.0.2.2/32, Prefix-SID: Index 2
Router-C: 192.0.2.3/32, Prefix-SID: Index 3
Router-D: 192.0.2.4/32, Prefix-SID: Index 4
```

```
      0                   1                   2                   3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |      Type     |     Length    |0|0|0|0|0|     |    RESERVED   |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |            Range = 4          |       32      |      192      |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |        0      |        2      |       1       |Prefix-SID Type|
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     | sub-TLV Length|     Flags     |   Algorithm   |               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                                             1 |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Example-2: If the following IPv4 prefixes need to be mapped into the corresponding Prefix-SID indexes:

```
10.1.1/24, Prefix-SID: Index 51
10.1.2/24, Prefix-SID: Index 52
10.1.3/24, Prefix-SID: Index 53
10.1.4/24, Prefix-SID: Index 54
10.1.5/24, Prefix-SID: Index 55
10.1.6/24, Prefix-SID: Index 56
10.1.7/24, Prefix-SID: Index 57
```

```
      0                   1                   2                   3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |      Type     |     Length    |0|0|0|0|0|     |    RESERVED   |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |            Range = 7          |       24      |      10       |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |        1      |        1      |Prefix-SID Type| sub-TLV Length|
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |    Flags      | Algorithm     |                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                            51 |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Example-3: If the following IPv6 prefixes need to be mapped into the corresponding Prefix-SID indexes:

```
2001:db8:1/48, Prefix-SID: Index 151
2001:db8:2/48, Prefix-SID: Index 152
2001:db8:3/48, Prefix-SID: Index 153
2001:db8:4/48, Prefix-SID: Index 154
```

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |1|0|0|0|0|     |    RESERVED   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Range = 4          |       48      |     0x20      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      0x01     |     0x0d      |      0xb8     |     0x00      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    0x01       |Prefix-SID Type| sub-TLV Length|  Flags        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Algorithm     |                    0                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      151      |
+-+-+-+-+-+-+-+-+
```

It is not expected that a network operator will be able to keep fully continuous Prefix / SID/Index mappings. In order to support noncontinuous mapping ranges an implementation MAY generate several instances of Binding TLVs.

For example if a router wants to advertise the following ranges:

Range 16: { 192.0.2.1-15, Index 1-15 }

Range 6: { 192.0.2.22-27, Index 22-27 }

Range 41: { 192.0.2.44-84, Index 80-120 }

A router would need to advertise three instances of the Binding TLV.

2.5. Multi-Topology SID/Label Binding TLV

The Multi-Topology SID/Label Binding TLV allows the support of M-ISIS as defined in [RFC5120]. The Multi-Topology SID/Label Binding TLV has the same format as the SID/Label Binding TLV defined in Section 2.4 with the difference consisting of a Multitopology Identifier (MTID) as defined here below:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |              MTID             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Flags     |   RESERVED    |             Range             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Prefix Length |          Prefix (variable)                   //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Sub-TLVs (variable)                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: Multi-Topology SID/Label Binding TLV format

where:

Type: 150

Length: variable

MTID is the multitopology identifier defined as:

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| RESVD |         MTID          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

RESVD: reserved bits. MUST be reset on transmission and ignored on receive.

MTID: a 12-bit field containing the non-zero ID of the topology being announced. The TLV MUST be ignored if the ID is zero. This is to ensure the consistent view of the standard unicast topology.

The other fields and Sub-TLVs are defined in Section 2.4.

3. Router Capabilities

This section defines sub-TLVs which are inserted into the IS-IS Router Capability TLV-242 that is defined in [RFC7981].

3.1. SR-Capabilities Sub-TLV

Segment Routing requires each router to advertise its SR data-plane capability and the range of MPLS label values it uses for Segment Routing in the case where global SIDs are allocated (i.e., global indexes). Data-plane capabilities and label ranges are advertised using the newly defined SR-Capabilities sub-TLV.

The Router Capability TLV specifies flags that control its advertisement. The SR Capabilities sub-TLV MUST be propagated throughout the level and MUST NOT be advertised across level boundaries. Therefore Router Capability TLV distribution flags are set accordingly, i.e., the S flag in the Router Capability TLV [RFC7981] MUST be unset.

The SR Capabilities sub-TLV has following format:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Type        |     Length    |    Flags      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Range                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                SID/Label Sub-TLV (variable)                 //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type: 2

Length: variable.

Flags: 1 octet of flags. The following are defined:

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|I|V|           |
+-+-+-+-+-+-+-+-+
```

where:

I-Flag: MPLS IPv4 flag. If set, then the router is capable of processing SR MPLS encapsulated IPv4 packets on all interfaces.

V-Flag: MPLS IPv6 flag. If set, then the router is capable of processing SR MPLS encapsulated IPv6 packets on all interfaces.

One or more SRGB Descriptor entries, each of which have the following format:

Range: 3 octets.

SID/Label sub-TLV (as defined in Section 2.3).

SID/Label sub-TLV contains the first value of the SRGB while the range contains the number of SRGB elements. The range value MUST be higher than 0.

The SR-Capabilities sub-TLV MAY be advertised in an LSP of any number but a router MUST NOT advertise more than one SR-Capabilities sub-TLV. A router receiving multiple SR-Capabilities sub-TLVs from the same originator SHOULD select the first advertisement in the lowest numbered LSP.

When multiple SRGB Descriptors are advertised the entries define an ordered set of ranges on which a SID index is to be applied. For this reason changing the order in which the descriptors are advertised will have a disruptive effect on forwarding.

When a router adds a new SRGB Descriptor to an existing SR-Capabilities sub-TLV the new Descriptor SHOULD add the newly configured block at the end of the sub-TLV and SHOULD NOT change the order of previously advertised blocks. Changing the order of the advertised descriptors will create label churn in the FIB and blackhole / misdirect some traffic during the IGP convergence. In particular, if a range which is not the last is extended it’s preferable to add a new range rather than extending the previously advertised range.

The originating router MUST ensure the order is unchanged after a graceful restart (using checkpointing, non-volatile storage or any other mechanism).

The originating router MUST NOT advertise overlapping ranges.

When a router receives multiple overlapping ranges, it MUST conform to the procedures defined in [I-D.ietf-spring-segment-routing-mpls].

Here follows an example of advertisement of multiple ranges:

The originating router advertises following ranges:

```
SR-Cap: range: 100, SID value: 100
SR-Cap: range: 100, SID value: 1000
SR-Cap: range: 100, SID value: 500
```

The receiving routers concatenate the ranges in the received order and build the SRGB as follows:

```
SRGB = [100, 199]
       [1000, 1099]
       [500, 599]
```

The indexes span multiple ranges:

```
index=0   means label 100
...
index 99  means label 199
index 100 means label 1000
index 199 means label 1099
...
index 200 means label 500
...
```

3.2. SR-Algorithm Sub-TLV

The router may use various algorithms when calculating reachability to other nodes or to prefixes attached to these nodes. Examples of these algorithms are metric based Shortest Path First (SPF), various sorts of Constrained SPF, etc. The SR-Algorithm sub-TLV allows the router to advertise the algorithms that the router is currently using. Algorithm values are defined in the "IGP Algorithm Type" registry defined in [I-D.ietf-ospf-segment-routing-extensions]. The following values have been defined:

0: Shortest Path First (SPF) algorithm based on link metric. This is the well-known shortest path algorithm as computed by the IS-IS Decision process. Consistent with the deployed practice for link-state protocols, algorithm 0 permits any node to overwrite the SPF path with a different path based on local policy.

1: Strict Shortest Path First (SPF) algorithm based on link metric. The algorithm is identical to algorithm 0 but algorithm 1 requires that all nodes along the path will honor the SPF routing decision. Local policy MUST NOT alter the forwarding decision computed by algorithm 1 at the node claiming to support algorithm 1.

The Router Capability TLV specifies flags that control its advertisement. The SR-Algorithm MUST be propagated throughout the level and MUST NOT be advertised across level boundaries. Therefore Router Capability TLV distribution flags are set accordingly, i.e., the S flag MUST be unset.

The SR-Algorithm sub-TLV is optional. It MUST NOT be advertsied more than once at a given level. A router receiving multiple SR-Algorithm sub-TLVs from the same originator SHOULD select the first advertisement in the lowest numbered LSP.

When the originating router does not advertise the SR-Algorithm sub-TLV, this implies that the only algorithm supported by routers supporting the extensions defined in this document is Algorithm 0.

When the originating router does advertise the SR-Algorithm sub-TLV, then algorithm 0 MUST be present while non-zero algorithms MAY be present.

The SR-Algorithm sub-TLV has the following format:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Type        |     Length    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Algorithm 1   | Algorithm 2   | Algorithm ... |  Algorithm n  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Type: 19

Length: variable.

Algorithm: 1 octet of algorithm

3.3. SR Local Block Sub-TLV

The SR Local Block (SRLB) Sub-TLV contains the range of labels the node has reserved for local SIDs. Local SIDs are used, e.g., for Adjacency-SIDs, and may also be allocated by components other than the IS-IS protocol. As an example, an application or a controller may instruct the router to allocate a specific local SID. Therefore, in order for such applications or controllers to know what are the local SIDs available in the router, it is required that the router advertises its SRLB.

The SRLB Sub-TLV is used for this purpose and has following format:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Type        |     Length    |    Flags      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Range                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                SID/Label Sub-TLV (variable)                 //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type: 22

Length: variable.

Flags: 1 octet of flags. None are defined at this stage.

One or more SRLB Descriptor entries, each of which have the following format:

Range: 3 octets.

SID/Label sub-TLV (as defined in Section 2.3).

SID/Label sub-TLV contains the first value of the SRLB while the range contains the number of SRLB elements. The range value MUST be higher than 0.

The SRLB sub-TLV MAY be advertised in an LSP of any number but a router MUST NOT advertise more than one SRLB sub-TLV. A router receiving multiple SRLB sub-TLVs, from the same originator, SHOULD select the first advertisement in the lowest numbered LSP.

The originating router MUST NOT advertise overlapping ranges.

When a router receives multiple overlapping ranges, it MUST conform to the procedures defined in [I-D.ietf-spring-segment-routing-mpls].

It is important to note that each time a SID from the SRLB is allocated, it should also be reported to all components (e.g.: controller or applications) in order for these components to have an up-to-date view of the current SRLB allocation and in order to avoid collision between allocation instructions.

Within the context of IS-IS, the reporting of local SIDs is done through IS-IS Sub-TLVs such as the Adjacency-SID. However, the reporting of allocated local SIDs may also be done through other means and protocols which are outside the scope of this document.

A router advertising the SRLB sub-TLV may also have other label ranges, outside the SRLB, for its local allocation purposes which are NOT advertised in the SRLB. For example, it is possible that an Adjacency-SID is allocated using a local label not part of the SRLB.

3.4. SRMS Preference Sub-TLV

The Segment Routing Mapping Server (SRMS) Preference sub-TLV is used in order to associate a preference with SRMS advertisements from a particular source.

The SRMS Preference sub-TLV has following format:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Type        |     Length    | Preference    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type: 24

Length: 1.

Preference: 1 octet. Unsigned 8 bit SRMS preference.

The SRMS Preference sub-TLV MAY be advertised in an LSP of any number but a router MUST NOT advertise more than one SRMS Preference sub-TLV. A router receiving multiple SRMS Preference sub-TLVs, from the same originator, SHOULD select the first advertisement in the lowest numbered LSP.

The use of the SRMS Preference during the SID selection process is described in [I-D.ietf-spring-segment-routing-ldp-interop]

4. IANA Considerations

This document requests allocation for the following TLVs and Sub-TLVs.

4.1. Sub TLVs for Type 22,23,25,141,222, and 223

This document makes the following registrations in the "sub-TLVs for TLV 22, 23, 25, 141, 222 and 223" registry.

Type	Description	22	23	25	141	222	223
31	Adjacency Segment Identifier	y	y	n	y	y	y
32	LAN Adjacency Segment Identifier	y	y	n	y	y	y

4.2. Sub TLVs for Type 135,235,236 and 237

This document makes the following registrations in the "sub-TLVs for TLV 135,235,236 and 237" registry.

Type	Description	135	235	236	237
3	Prefix Segment Identifier	y	y	y	y

4.3. Sub TLVs for Type 242

This document makes the following registrations in the "sub-TLVs for TLV 242" registry.

Type	Description
2	Segment Routing Capability
19	Segment Routing Algorithm
22	Segment Routing Local Block (SRLB)
24	Segment Routing Mapping Server Preference (SRMS Preference)

4.4. New TLV Codepoint and Sub-TLV registry

This document registers the following TLV:

Value	Name	IIH	LSP	SNP	Purge
149	Segment Identifier/Label Binding	n	y	n	n
150	Multi-Topology Segment Identifier /Label Binding	n	y	n	n

This document creates the following sub-TLV Registry:

Name: sub-TLVs for TLVs 149 and 150
Registration Procedure: Expert Review

Type	Description
0	Reserved
1	SID/Label
2	Unassigned
3	Prefix SID
4-255	Unassigned

5. Security Considerations

With the use of the extensions defined in this document, IS-IS carries information which will be used to program the MPLS data plane [RFC3031]. In general, the same types of attacks that can be carried out on the IP/IPv6 control plane can be carried out on the MPLS control plane resulting in traffic being misrouted in the respective data planes. However, the latter may be more difficult to detect and isolate.

Existing security extensions as described in [RFC5304] and [RFC5310] apply to these segment routing extensions.

6. Acknowledgements

We would like to thank Dave Ward, Dan Frost, Stewart Bryant, Pierre Francois and Jesper Skrivers for their contribution to the content of this document.

7. Contributors

The following people gave a substantial contribution to the content of this document and should be considered as co-authors:

Stephane Litkowski
Orange
FR

Email: stephane.litkowski@orange.com

Jeff Tantsura
Apstra, Inc.

Email: jefftant@gmail.com

Peter Psenak

Cisco Systems Inc.
US

Email: ppsenak@cisco.com

Martin Horneffer
Deutsche Telekom
DE

Email: Martin.Horneffer@telekom.de

Wim Henderickx
Nokia
BE

Email: wim.henderickx@nokia.com

Edward Crabbe
Oracle
US

Email: edward.crabbe@oracle.com

Rob Shakir
Google
UK

Email: robjs@google.com

Igor Milojevic
Individual
RS

Email: milojevicigor@gmail.com

Saku Ytti
TDC
FI

Email: saku@ytti.fi

Steven Luong
Cisco Systems Inc.

US

Email: sluong@cisco.com

8. References

8.1. Normative References

[I-D.ietf-ospf-segment-routing-extensions]
Psenak, P., Previdi, S., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", draft-ietf-ospf-segment-routing-extensions-27 (work in progress), December 2018.

[I-D.ietf-spring-segment-routing-ldp-interop]
Bashandy, A., Filsfils, C., Previdi, S., Decraene, B., and S. Litkowski, "Segment Routing interworking with LDP", draft-ietf-spring-segment-routing-ldp-interop-15 (work in progress), September 2018.

[I-D.ietf-spring-segment-routing-mpls]
Bashandy, A., Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-22 (work in progress), May 2019.

[ISO10589]
International Organization for Standardization, "Intermediate system to Intermediate system intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473)", ISO/IEC 10589:2002, Second Edition, Nov 2002.

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

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <https://www.rfc-editor.org/info/rfc3031>.

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

[RFC5304] Li, T. and R. Atkinson, "IS-IS Cryptographic Authentication", RFC 5304, DOI 10.17487/RFC5304, October 2008, <https://www.rfc-editor.org/info/rfc5304>.

[RFC5310] Bhatia, M., Manral, V., Li, T., Atkinson, R., White, R., and M. Fanto, "IS-IS Generic Cryptographic Authentication", RFC 5310, DOI 10.17487/RFC5310, February 2009, <https://www.rfc-editor.org/info/rfc5310>.

[RFC7794] Ginsberg, L., Ed., Decraene, B., Previdi, S., Xu, X., and U. Chunduri, "IS-IS Prefix Attributes for Extended IPv4 and IPv6 Reachability", RFC 7794, DOI 10.17487/RFC7794, March 2016, <https://www.rfc-editor.org/info/rfc7794>.

[RFC7981] Ginsberg, L., Previdi, S., and M. Chen, "IS-IS Extensions for Advertising Router Information", RFC 7981, DOI 10.17487/RFC7981, October 2016, <https://www.rfc-editor.org/info/rfc7981>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

8.2. Informative References

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <https://www.rfc-editor.org/info/rfc5305>.

[RFC5308] Hopps, C., "Routing IPv6 with IS-IS", RFC 5308, DOI 10.17487/RFC5308, October 2008, <https://www.rfc-editor.org/info/rfc5308>.

[RFC5311] McPherson, D., Ed., Ginsberg, L., Previdi, S., and M. Shand, "Simplified Extension of Link State PDU (LSP) Space for IS-IS", RFC 5311, DOI 10.17487/RFC5311, February 2009, <https://www.rfc-editor.org/info/rfc5311>.

[RFC5316] Chen, M., Zhang, R., and X. Duan, "ISIS Extensions in Support of Inter-Autonomous System (AS) MPLS and GMPLS Traffic Engineering", RFC 5316, DOI 10.17487/RFC5316, December 2008, <https://www.rfc-editor.org/info/rfc5316>.

[RFC7855] Previdi, S., Ed., Filsfils, C., Ed., Decraene, B., Litkowski, S., Horneffer, M., and R. Shakir, "Source Packet Routing in Networking (SPRING) Problem Statement and Requirements", RFC 7855, DOI 10.17487/RFC7855, May 2016, <https://www.rfc-editor.org/info/rfc7855>.

Authors' Addresses

Stefano Previdi (editor)
Huawei
IT

Email: stefano@previdi.net

Les Ginsberg (editor)
Cisco Systems, Inc.
USA

Email: ginsberg@cisco.com

Clarence Filsfils
Cisco Systems, Inc.
Brussels
BE

Email: cfilsfil@cisco.com

Ahmed Bashandy
Arrcus

Email: abashandy.ietf@gmail.com

Hannes Gredler
RtBrick Inc.

Email: hannes@rtbrick.com

Bruno Decraene
Orange
FR

Email: bruno.decraene@orange.com

IS-IS Working Group S. Litkowski
Internet-Draft Cisco Systems
Intended status: Standards Track Y. Qu
Expires: January 13, 2021 Futurewei
P. Sarkar
Individual
I. Chen
The MITRE Corporation
J. Tantsura
Apstra
July 12, 2020

YANG Data Model for IS-IS Segment Routing
draft-ietf-isis-sr-yang-08

Abstract

This document defines a YANG data model that can be used to configure and manage IS-IS Segment Routing.

Status of This Memo

Copyright Notice

Table of Contents

1. Overview

YANG [RFC6020] [RFC7950] is a data definition language used to define the contents of a conceptual data store that allows networked devices to be managed using NETCONF [RFC6241]. YANG is proving relevant beyond its initial confines, as bindings to other interfaces (e.g., ReST) and encodings other than XML (e.g., JSON) are being defined. Furthermore, YANG data models can be used as the basis for implementation of other interfaces, such as CLI and programmatic APIs.

This document defines a YANG data model that can be used to configure and manage IS-IS Segment Routing [RFC8667] and it is an augmentation to the IS-IS YANG data model.

The YANG modules in this document conform to the Network Management Datastore Architecture (NMDA) [RFC8342].

2. Requirements Language

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

3. Tree Diagrams

This document uses the graphical representation of data models defined in [RFC8340].

4. IS-IS Segment Routing

This document defines a model for IS-IS Segment Routing feature. It is an augmentation of the IS-IS base model.

The IS-IS SR YANG module requires support for the base segment routing module [I-D.ietf-spring-sr-yang], which defines the global segment routing configuration independent of any specific routing protocol configuration, and support of IS-IS base model [I-D.ietf-isis-yang-isis-cfg] which defines basic IS-IS configuration and state.

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

```
module: ietf-isis-sr
  augment /rt:routing/rt:control-plane-protocols
          /rt:control-plane-protocol/isis:isis:
    +--rw segment-routing
    |  +--rw enabled?    boolean
    |  +--rw bindings
    |     +--rw advertise
    |     |  +--rw policies*   string
    |     +--rw receive?    boolean
    +--rw protocol-srgb {sr-mpls:protocol-srgb}?
       +--rw srgb* [lower-bound upper-bound]
          +--rw lower-bound    uint32
          +--rw upper-bound    uint32
  augment /rt:routing/rt:control-plane-protocols
          /rt:control-plane-protocol/isis:isis/isis:interfaces
          /isis:interface:
    +--rw segment-routing
```

```
       +--rw adjacency-sid
          +--rw adj-sids* [value]
          |  +--rw value-type?   enumeration
          |  +--rw value         uint32
          |  +--rw protected?    boolean
          +--rw advertise-adj-group-sid* [group-id]
          |  +--rw group-id    uint32
          +--rw advertise-protection?      enumeration
  augment /rt:routing/rt:control-plane-protocols
          /rt:control-plane-protocol/isis:isis/isis:interfaces
          /isis:interface/isis:fast-reroute:
    +--rw ti-lfa {ti-lfa}?
       +--rw enable?   boolean
  augment /rt:routing/rt:control-plane-protocols
          /rt:control-plane-protocol/isis:isis/isis:interfaces
          /isis:interface/isis:fast-reroute/isis:lfa/isis:remote-lfa:
    +--rw use-segment-routing-path?   boolean {remote-lfa-sr}?
  augment /rt:routing/rt:control-plane-protocols
          /rt:control-plane-protocol/isis:isis/isis:interfaces
          /isis:interface/isis:adjacencies/isis:adjacency:
    +--ro adjacency-sid* [value]
       +--ro af?                      iana-rt-types:address-family
       +--ro value                    uint32
       +--ro weight?                  uint8
       +--ro protection-requested?   boolean
  augment /rt:routing/rt:control-plane-protocols
          /rt:control-plane-protocol/isis:isis/isis:database
          /isis:levels/isis:lsp/isis:router-capabilities:
    +--ro sr-capability
    |  +--ro sr-capability
    |  |  +--ro sr-capability-bits*   identityref
    |  +--ro global-blocks
    |     +--ro global-block* []
    |        +--ro range-size?    uint32
    |        +--ro sid-sub-tlv
    |           +--ro sid?   uint32
    +--ro sr-algorithms
    |  +--ro sr-algorithm*   uint8
    +--ro local-blocks
    |  +--ro local-block* []
    |     +--ro range-size?    uint32
    |     +--ro sid-sub-tlv
    |        +--ro sid?   uint32
    +--ro srms-preference
       +--ro preference?   uint8
  augment /rt:routing/rt:control-plane-protocols
          /rt:control-plane-protocol/isis:isis/isis:database/isis:levels
          /isis:lsp/isis:extended-is-neighbor/isis:neighbor:
```

```
    +--ro sid-list* [value]
       +--ro adj-sid-flags
       |  +--ro bits*   identityref
       +--ro weight?          uint8
       +--ro neighbor-id?     isis:system-id
       +--ro value            uint32
  augment /rt:routing/rt:control-plane-protocols
          /rt:control-plane-protocol/isis:isis/isis:database
          /isis:levels/isis:lsp/isis:mt-is-neighbor/isis:neighbor:
    +--ro sid-list* [value]
       +--ro adj-sid-flags
       |  +--ro bits*   identityref
       +--ro weight?          uint8
       +--ro neighbor-id?     isis:system-id
       +--ro value            uint32
  augment /rt:routing/rt:control-plane-protocols
          /rt:control-plane-protocol/isis:isis/isis:database
          /isis:levels/isis:lsp/isis:extended-ipv4-reachability
          /isis:prefixes:
    +--ro sid-list* [value]
       +--ro perfix-sid-flags
       |  +--ro bits*   identityref
       +--ro algorithm?          uint8
       +--ro value               uint32
  augment /rt:routing/rt:control-plane-protocols
          /rt:control-plane-protocol/isis:isis/isis:database
          /isis:levels/isis:lsp/isis:mt-extended-ipv4-reachability
          /isis:prefixes:
    +--ro sid-list* [value]
       +--ro perfix-sid-flags
       |  +--ro bits*   identityref
       +--ro algorithm?          uint8
       +--ro value               uint32
  augment /rt:routing/rt:control-plane-protocols
          /rt:control-plane-protocol/isis:isis/isis:database
          /isis:levels/isis:lsp/isis:ipv6-reachability/isis:prefixes:
    +--ro sid-list* [value]
       +--ro perfix-sid-flags
       |  +--ro bits*   identityref
       +--ro algorithm?          uint8
       +--ro value               uint32
  augment /rt:routing/rt:control-plane-protocols
          /rt:control-plane-protocol/isis:isis/isis:database
          /isis:levels/isis:lsp/isis:mt-ipv6-reachability/isis:prefixes:
    +--ro sid-list* [value]
       +--ro perfix-sid-flags
       |  +--ro bits*   identityref
       +--ro algorithm?          uint8
```

```
       +--ro value               uint32
  augment /rt:routing/rt:control-plane-protocols
          /rt:control-plane-protocol/isis:isis/isis:database
          /isis:levels/isis:lsp:
    +--ro segment-routing-bindings* [fec range]
       +--ro fec                  string
       +--ro range                uint16
       +--ro sid-binding-flags
       |  +--ro bits*   identityref
       +--ro binding
          +--ro prefix-sid
             +--ro sid-list* [value]
                +--ro perfix-sid-flags
                |  +--ro bits*   identityref
                +--ro algorithm?          uint8
                +--ro value               uint32
```

5. IS-IS Segment Routing configuration

5.1. Segment Routing activation

Activation of segment-routing IS-IS is done by setting the "enable" leaf to true. This triggers advertisement of segment-routing extensions based on the configuration parameters that have been setup using the base segment routing module.

5.2. Advertising mapping server policy

The base segment routing module defines mapping server policies. By default, IS-IS will not advertise nor receive any mapping server entry. The IS-IS segment-routing module allows to advertise one or multiple mapping server policies through the "bindings/advertise/policies" leaf-list. The "bindings/receive" leaf allows to enable the reception of mapping server entries.

5.3. IP Fast reroute

IS-IS SR model augments the fast-reroute container under interface. It brings the ability to activate TI-LFA (topology independent LFA) and also enhances remote LFA to use segment-routing tunneling instead of LDP.

6. IS-IS Segment Routing YANG Module

```
<CODE BEGINS> file "ietf-isis-sr@2020-07-12.yang"
module ietf-isis-sr {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:"
```

```
             + "yang:ietf-isis-sr";
     prefix isis-sr;


     import ietf-routing {
       prefix "rt";
       reference "RFC 8349 - A YANG Data Model for Routing
                  Management (NMDA Version)";
     }

     import ietf-segment-routing-common {
       prefix "sr-cmn";
     }

     import ietf-segment-routing-mpls {
       prefix "sr-mpls";
     }

     import ietf-isis {
       prefix "isis";
     }

     import iana-routing-types {
       prefix "iana-rt-types";
       reference "RFC 8294 - Common YANG Data Types for the
                  Routing Area";
     }

     organization
      "IETF LSR - LSR Working Group";

     contact
       "WG List:  <mailto:lsr@ietf.org>

       Editor:    Stephane Litkowski
                  <mailto:stephane.litkowski@orange.com>

       Author:    Acee Lindem
                  <mailto:acee@cisco.com>
       Author:    Yingzhen Qu
                  <mailto:yingzhen.qu@futurewei.com>
       Author:    Pushpasis Sarkar
                  <mailto:pushpasis.ietf@gmail.com>
       Author:    Ing-Wher Chen
                  <mailto:ingwherchen@mitre.org>
       Author:    Jeff Tantsura
                  <mailto:jefftant.ietf@gmail.com>
       ";
```

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

       This YANG model conforms to the Network Management
       Datastore Architecture (NMDA) as described in RFC 8242.

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

       Redistribution and use in source and binary forms, with or
       without modification, is permitted pursuant to, and subject to
       the license terms contained in, the Simplified BSD License set
       forth in Section 4.c of the IETF Trust's Legal Provisions
       Relating to IETF Documents
       (https://trustee.ietf.org/license-info).

       This version of this YANG module is part of RFC XXXX
       (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
       for full legal notices.

       The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL
       NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'NOT RECOMMENDED',
       'MAY', and 'OPTIONAL' in this document are to be interpreted as
       described in BCP 14 (RFC 2119) (RFC 8174) when, and only when,
       they appear in all capitals, as shown here.

       This version of this YANG module is part of RFC XXXX;
       see the RFC itself for full legal notices.";

     reference "RFC XXXX";

     revision 2020-07-12 {
     description
       "Initial revision.";
     reference "RFC XXXX";
     }


     /* Identities */
     identity sr-capability {
       description
         "Base identity for ISIS SR-Capabilities sub-TLV flgs";
     }

     identity mpls-ipv4 {
       base sr-capability;
```

```
         description
           "If set, then the router is capable of
            processing SR MPLS encapsulated IPv4 packets
            on all interfaces.";
       }

       identity mpls-ipv6 {
         base sr-capability;
         description
           "If set, then the router is capable of
            processing SR MPLS encapsulated IPv6 packets
            on all interfaces.";
       }

       identity prefix-sid-bit {
         description
           "Base identity for prefix sid sub-tlv bits.";
       }

       identity r-bit {
         base prefix-sid-bit;
         description
          "Re-advertisement Flag.";
       }

       identity n-bit {
         base prefix-sid-bit;
         description
          "Node-SID Flag.";
       }

       identity p-bit {
         base prefix-sid-bit;
         description
          "No-PHP (No Penultimate Hop-Popping) Flag.";
       }

       identity e-bit {
         base prefix-sid-bit;
         description
          "Explicit NULL Flag.";
       }

       identity v-bit {
         base prefix-sid-bit;
         description
          "Value Flag.";
       }
```

```
     identity l-bit {
       base prefix-sid-bit;
       description
        "Local Flag.";
     }

     identity adj-sid-bit {
       description
         "Base identity for adj sid sub-tlv bits.";
     }

     identity f-bit {
       base adj-sid-bit;
       description
         "Address-Family flag.";
     }

     identity b-bit {
       base adj-sid-bit;
       description
         "Backup flag.";
     }

     identity vi-bit {
       base adj-sid-bit;
       description
         "Value/Index flag.";
     }

     identity lo-bit {
       base adj-sid-bit;
       description
         "Local flag.";
     }

     identity s-bit {
       base adj-sid-bit;
       description
         "Group flag.";
     }

     identity pe-bit {
       base adj-sid-bit;
       description
         "Persistent flag.";
     }

     identity sid-binding-bit {
```

```
       description
         "Base identity for sid binding tlv bits.";
     }

     identity af-bit {
       base sid-binding-bit;
       description
         "Address-Family flag.";
     }

     identity m-bit {
       base sid-binding-bit;
       description
         "Mirror Context flag.";
     }

     identity sf-bit {
       base sid-binding-bit;
       description
         "S flag. If set, the binding label tlv should be flooded
          across the entire routing domain.";
     }

     identity d-bit {
       base sid-binding-bit;
       description
         "Leaking flag.";
     }

     identity a-bit {
       base sid-binding-bit;
       description
         "Attached flag.";
     }

     /* Features */

     feature remote-lfa-sr {
       description
        "Enhance rLFA to use SR path.";
     }

     feature ti-lfa {
       description
        "Enhance IPFRR with ti-lfa
        support";
     }
```

```
    /* Groupings */

    grouping sid-sub-tlv {
      description "SID/Label sub-TLV grouping.";
      container sid-sub-tlv {
        description
          "Used to advertise the SID/Label associated with a
        prefix or adjacency.";
        leaf sid {
          type uint32;
        description
          "Segment Identifier (SID) - A 20 bit label or
           32 bit SID.";
        }
      }
    }

    grouping sr-capability {
      description
        "SR capability grouping.";
      container sr-capability {
        description
          "Segment Routing capability.";
        container sr-capability {
          leaf-list sr-capability-bits {
            type identityref {
              base sr-capability;
            }
            description "SR Capbility sub-tlv flags list.";
          }
          description
            "SR Capability Flags.";
        }
        container global-blocks {
          description
            "Segment Routing Global Blocks.";
          list global-block {
            description "Segment Routing Global Block.";
            leaf range-size {
              type uint32;
              description "The SID range.";
            }
            uses sid-sub-tlv;
          }
        }
      }
    }
```

```
     grouping sr-algorithm {
       description
         "SR algorithm grouping.";
       container sr-algorithms {
         description "All SR algorithms.";
         leaf-list sr-algorithm {
           type uint8;
           description
             "The Segment Routing (SR) algorithms that the router is
              currently using.";
         }
       }
     }

     grouping srlb {
       description
         "SR Local Block grouping.";
       container local-blocks {
         description "List of SRLBs.";
         list local-block {
           description "Segment Routing Local Block.";
           leaf range-size {
             type uint32;
             description "The SID range.";
           }
           uses sid-sub-tlv;
         }
       }
     }

     grouping srms-preference {
       description "The SRMS preference TLV is used to advertise
                    a preference associated with the node that acts
                    as an SR Mapping Server.";
       container srms-preference {
         description "SRMS Preference TLV.";
         leaf preference {
           type uint8 {
             range "0 .. 255";
           }
           description "SRMS preference TLV, vlaue from 0 to 255.";
         }
       }
     }

     grouping adjacency-state {
       description
         "This group will extend adjacency state.";
```

```
       list adjacency-sid {
         key value;
         config false;
         leaf af {
           type iana-rt-types:address-family;
           description
             "Address-family associated with the
              segment ID";
         }
         leaf value {
           type uint32;
           description
             "Value of the Adj-SID.";
         }
         leaf weight {
           type uint8;
           description
             "Weight associated with
             the adjacency SID.";
         }
         leaf protection-requested {
           type boolean;
           description
             "Describe if the adjacency SID
              must be protected.";
         }
         description
           "List of adjacency Segment IDs.";
       }
     }

     grouping prefix-segment-id {
       description
         "This group defines segment routing extensions
          for prefixes.";

       list sid-list {
         key value;

         container perfix-sid-flags {
           leaf-list bits {
             type identityref {
               base prefix-sid-bit;
             }
             description
               "Prefix SID Sub-TLV flag bits list.";
           }
           description
```

```
             "Describes flags associated with the
              segment ID.";
         }

         leaf algorithm {
           type uint8;
           description
             "Algorithm to be used for path computation.";
         }
         leaf value {
           type uint32;
           description
            "Value of the prefix-SID.";
         }
         description
           "List of segments.";
       }
     }

     grouping adjacency-segment-id {
       description
         "This group defines segment routing extensions
          for adjacencies.";

       list sid-list {
         key value;

         container adj-sid-flags {
           leaf-list bits {
             type identityref {
               base adj-sid-bit;
             }
             description "Adj sid sub-tlv flags list.";
             }
           description "Adj-sid sub-tlv flags.";
         }

         leaf weight {
           type uint8;
           description
             "The value represents the weight of the Adj-SID
              for the purpose of load balancing.";
         }
         leaf neighbor-id {
           type isis:system-id;
           description
             "Describes the system ID of the neighbor
              associated with the SID value. This is only
```

```
              used on LAN adjacencies.";
         }
         leaf value {
           type uint32;
           description
             "Value of the Adj-SID.";
         }
         description
           "List of segments.";
       }
     }

     grouping segment-routing-binding-tlv {
       list segment-routing-bindings {
         key "fec range";

         leaf fec {
           type string;
           description
           "IP (v4 or v6) range to be bound to SIDs.";
         }

         leaf range {
           type uint16;
           description
             "Describes number of elements to assign
              a binding to.";
         }

         container sid-binding-flags {
           leaf-list bits {
             type identityref {
               base sid-binding-bit;
             }
             description
               "SID Binding TLV flag bits list.";
           }
           description
             "Binding flags.";
         }

         container binding {
           container prefix-sid {
             uses prefix-segment-id;
             description
               "Binding prefix SID to the range.";
           }
           description
```

```
             "Bindings associated with the range.";
         }

         description
           "This container describes list of SID/Label bindings.
            ISIS reference is TLV 149.";
       }
       description
         "Defines binding TLV for database.";
     }

     /* Cfg */

     augment "/rt:routing/" +
             "rt:control-plane-protocols/rt:control-plane-protocol"+
             "/isis:isis" {
       when "/rt:routing/rt:control-plane-protocols/"+
            "rt:control-plane-protocol/rt:type = 'isis:isis'" {
         description
           "This augment ISIS routing protocol when used";
       }
       description
         "This augments ISIS protocol configuration
          with segment routing.";

       uses sr-mpls:sr-controlplane;
       container protocol-srgb {
         if-feature sr-mpls:protocol-srgb;
         uses sr-cmn:srgb;
         description
           "Per-protocol SRGB.";
       }
     }

     augment "/rt:routing/" +
             "rt:control-plane-protocols/rt:control-plane-protocol"+
             "/isis:isis/isis:interfaces/isis:interface" {
       when "/rt:routing/rt:control-plane-protocols/"+
            "rt:control-plane-protocol/rt:type = 'isis:isis'" {
         description
           "This augment ISIS routing protocol when used";
       }
       description
         "This augments ISIS protocol configuration
          with segment routing.";

       uses sr-mpls:igp-interface;
     }
```

```
     augment "/rt:routing/" +
             "rt:control-plane-protocols/rt:control-plane-protocol"+
             "/isis:isis/isis:interfaces/isis:interface"+
             "/isis:fast-reroute" {
       when "/rt:routing/rt:control-plane-protocols/"+
            "rt:control-plane-protocol/rt:type = 'isis:isis'" {
         description
           "This augment ISIS routing protocol when used";
       }
       description
         "This augments ISIS IP FRR with TILFA.";

       container ti-lfa {
         if-feature ti-lfa;
         leaf enable {
           type boolean;
           description
             "Enables TI-LFA computation.";
         }
         description
           "TILFA configuration.";
       }
     }

     augment "/rt:routing/" +
             "rt:control-plane-protocols/rt:control-plane-protocol"+
             "/isis:isis/isis:interfaces/isis:interface"+
             "/isis:fast-reroute/isis:lfa/isis:remote-lfa" {
       when "/rt:routing/rt:control-plane-protocols/"+
            "rt:control-plane-protocol/rt:type = 'isis:isis'" {
         description
           "This augment ISIS routing protocol when used";
       }
       description
         "This augments ISIS remoteLFA config with
          use of segment-routing path.";

       leaf use-segment-routing-path {
         if-feature remote-lfa-sr;
         type boolean;
         description
           "force remote LFA to use segment routing
            path instead of LDP path.";
       }
     }

     /* Operational states */
```

```
     augment "/rt:routing/" +
             "rt:control-plane-protocols/rt:control-plane-protocol"+
             "/isis:isis/isis:interfaces/isis:interface" +
             "/isis:adjacencies/isis:adjacency" {
       when "/rt:routing/rt:control-plane-protocols/"+
            "rt:control-plane-protocol/rt:type = 'isis:isis'" {
         description
           "This augment ISIS routing protocol when used";
       }
       description
         "This augments ISIS protocol configuration
          with segment routing.";

       uses adjacency-state;
     }

     augment "/rt:routing/" +
             "rt:control-plane-protocols/rt:control-plane-protocol"+
             "/isis:isis/isis:database/isis:levels/isis:lsp"+
             "/isis:router-capabilities" {
       when "/rt:routing/rt:control-plane-protocols/"+
            "rt:control-plane-protocol/rt:type = 'isis:isis'" {
         description
           "This augment ISIS routing protocol when used";
       }
       description
         "This augments ISIS protocol LSDB router capability.";

       uses sr-capability;
       uses sr-algorithm;
       uses srlb;
       uses srms-preference;
     }

     augment "/rt:routing/" +
             "rt:control-plane-protocols/rt:control-plane-protocol"+
             "/isis:isis/isis:database/isis:levels/isis:lsp"+
             "/isis:extended-is-neighbor/isis:neighbor" {
       when "/rt:routing/rt:control-plane-protocols/"+
            "rt:control-plane-protocol/rt:type = 'isis:isis'" {
         description
           "This augment ISIS routing protocol when used";
       }
       description
         "This augments ISIS protocol LSDB neighbor.";
          uses adjacency-segment-id;
     }
```

```
     augment "/rt:routing/" +
             "rt:control-plane-protocols/rt:control-plane-protocol"+
             "/isis:isis/isis:database/isis:levels/isis:lsp"+
             "/isis:mt-is-neighbor/isis:neighbor" {
       when "/rt:routing/rt:control-plane-protocols/"+
            "rt:control-plane-protocol/rt:type = 'isis:isis'" {
         description
           "This augment ISIS routing protocol when used";
       }
       description
         "This augments ISIS protocol LSDB neighbor.";
          uses adjacency-segment-id;
     }

     augment "/rt:routing/" +
             "rt:control-plane-protocols/rt:control-plane-protocol"+
             "/isis:isis/isis:database/isis:levels/isis:lsp"+
             "/isis:extended-ipv4-reachability/isis:prefixes" {
       when "/rt:routing/rt:control-plane-protocols/"+
            "rt:control-plane-protocol/rt:type = 'isis:isis'" {
         description
           "This augment ISIS routing protocol when used";
       }
       description
         "This augments ISIS protocol LSDB prefix.";
          uses prefix-segment-id;
     }

     augment "/rt:routing/" +
             "rt:control-plane-protocols/rt:control-plane-protocol"+
             "/isis:isis/isis:database/isis:levels/isis:lsp"+
             "/isis:mt-extended-ipv4-reachability/isis:prefixes" {
       when "/rt:routing/rt:control-plane-protocols/"+
            "rt:control-plane-protocol/rt:type = 'isis:isis'" {
         description
           "This augment ISIS routing protocol when used";
       }
       description
         "This augments ISIS protocol LSDB prefix.";
       uses prefix-segment-id;
     }

     augment "/rt:routing/" +
             "rt:control-plane-protocols/rt:control-plane-protocol"+
             "/isis:isis/isis:database/isis:levels/isis:lsp"+
             "/isis:ipv6-reachability/isis:prefixes" {
       when "/rt:routing/rt:control-plane-protocols/"+
            "rt:control-plane-protocol/rt:type = 'isis:isis'" {
```

```
          description
            "This augment ISIS routing protocol when used";
        }
        description
          "This augments ISIS protocol LSDB prefix.";
        uses prefix-segment-id;
      }

      augment "/rt:routing/" +
              "rt:control-plane-protocols/rt:control-plane-protocol"+
              "/isis:isis/isis:database/isis:levels/isis:lsp"+
              "/isis:mt-ipv6-reachability/isis:prefixes" {
        when "/rt:routing/rt:control-plane-protocols/"+
             "rt:control-plane-protocol/rt:type = 'isis:isis'" {
          description
            "This augment ISIS routing protocol when used";
        }
        description
          "This augments ISIS protocol LSDB prefix.";
        uses prefix-segment-id;
      }

      augment "/rt:routing/" +
              "rt:control-plane-protocols/rt:control-plane-protocol"+
              "/isis:isis/isis:database/isis:levels/isis:lsp" {
        when "/rt:routing/rt:control-plane-protocols/"+
             "rt:control-plane-protocol/rt:type = 'isis:isis'" {
          description
            "This augment ISIS routing protocol when used";
        }
        description
          "This augments ISIS protocol LSDB.";
        uses segment-routing-binding-tlv;
      }


      /* Notifications */

    }
    <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.

8. Contributors

Authors would like to thank Derek Yeung, Acee Lindem, Yi Yang for their major contributions to the draft.

9. Acknowledgements

MITRE has approved this document for Public Release, Distribution Unlimited, with Public Release Case Number 19-3033.

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

11. Change log for ietf-isis-sr YANG module

11.1. From version -03 to version -04

- o Fixed yang module indentations.

11.2. From version -02 to version -03

- o Change address-family type according to routing types.

11.3. From isis-sr document version -01 to version -02

- NMDA compliancy.
- Added SRLB in configuration and LSDB.
- Added SR capability in LSDB.
- Added SR algorithms in LSDB.
- Added SRMS preference in LSDB.
- Alignment with iana-rt-types module.
- Align binding SID with draft-ietf-isis-segment-routing-extensions-13.

11.4. From isis-sr document version -00 to version -01

- Added P-Flag in Adj-SID.

11.5. From isis document version -12 to isis-sr document version -00

- Separate document for IS-IS SR extensions.

11.6. From isis document version -12 to version -13

- Align with new segment routing common module.

11.7. From isis document version -09 to version -11

- Fixed XPATH in 'when' expressions.

11.8. From isis document version -08 to version -09

- Align to draft-ietf-netmod-routing-cfg-23.

11.9. From isis document version -07 to version -08

- Align to draft-ietf-netmod-routing-cfg-21.

12. Normative References

[I-D.ietf-isis-yang-isis-cfg]
Litkowski, S., Yeung, D., Lindem, A., Zhang, Z., and L. Lhotka, "YANG Data Model for IS-IS Protocol", draft-ietf-isis-yang-isis-cfg-42 (work in progress), October 2019.

[I-D.ietf-spring-sr-yang]
Litkowski, S., Qu, Y., Sarkar, P., and J. Tantsura, "YANG Data Model for Segment Routing", draft-ietf-spring-sr-yang-15 (work in progress), December 2017.

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

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

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

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

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

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <https://www.rfc-editor.org/info/rfc8340>.

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <https://www.rfc-editor.org/info/rfc8342>.

[RFC8667] Previdi, S., Ed., Ginsberg, L., Ed., Filsfils, C., Bashandy, A., Gredler, H., and B. Decraene, "IS-IS Extensions for Segment Routing", RFC 8667, DOI 10.17487/RFC8667, December 2019, <https://www.rfc-editor.org/info/rfc8667>.

Authors' Addresses

Stephane Litkowski
Cisco Systems

Email: slitkows.ietf@gmail.com

Yinghzen Qu
Futurewei

Email: yingzhen.qu@futurewei.com

Pushpasis Sarkar
Individual

Email: pushpasis.ietf@gmail.com

Ing-Wher Chen
The MITRE Corporation

Email: ingwherchen@mitre.org

Jeff Tantsura
Apstra

Email: jefftant.ietf@gmail.com

IS-IS Working Group S. Litkowski
Internet-Draft Cisco Systems
Intended status: Standards Track D. Yeung
Expires: April 17, 2020 Arrcus, Inc
A. Lindem
Cisco Systems
J. Zhang
Juniper Networks
L. Lhotka
CZ.NIC
October 15, 2019

YANG Data Model for IS-IS Protocol
draft-ietf-isis-yang-isis-cfg-42

Abstract

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

Requirements Language

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

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

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

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

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

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

2. Design of the Data Model

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

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

```
module: ietf-isis
augment /rt:routing/rt:ribs/rt:rib/rt:routes/rt:route:
  +--ro metric?       uint32
  +--ro tag*          uint64
  +--ro route-type?   enumeration
augment /if:interfaces/if:interface:
  +--rw clns-mtu?   uint16 {osi-interface}?
augment /rt:routing/rt:control-plane-protocols/rt:
            control-plane-protocol:
  +--rw isis
     +--rw 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 lsp-mtu?                 uint16
     +--rw lsp-lifetime?            uint16
     +--rw lsp-refresh?             rt-types:timer-value-seconds16
     |                                  {lsp-refresh}?
     +--rw poi-tlv?                 boolean {poi-tlv}?
     +--rw graceful-restart {graceful-restart}?
     |  +--rw 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 metric-type
     |  +--rw value?     enumeration
     |  +--rw level-1
     |  |     ...
     |  +--rw level-2
     |        ...
     +--rw default-metric
     |  +--rw value?     wide-metric
     |  +--rw level-1
     |  |     ...
     |  +--rw level-2
     |        ...
     +--rw auto-cost {auto-cost}?
     |  +--rw enable?                 boolean
     |  +--rw reference-bandwidth?   uint32
     +--rw authentication
     |  +--rw (authentication-type)?
     |  |     ...
     |  +--rw level-1
     |  |     ...
     |  +--rw level-2
     |        ...
     +--rw address-families {nlpid-control}?
     |  +--rw address-family-list* [address-family]
     |        ...
     +--rw mpls
     |  +--rw te-rid {te-rid}?
     |  |     ...
     |  +--rw ldp
     |        ...
     +--rw spf-control
     |  +--rw paths?            uint16 {max-ecmp}?
     |  +--rw ietf-spf-delay {ietf-spf-delay}?
     |        ...
     +--rw fast-reroute {fast-reroute}?
     |  +--rw lfa {lfa}?
     +--rw preference
     |  +--rw (granularity)?
     |        ...
     +--rw overload
     |  +--rw status?   boolean
     +--rw overload-max-metric {overload-max-metric}?
     |  +--rw timeout?   rt-types:timer-value-seconds16
     +--ro spf-log
     |  +--ro event* [id]
     |        ...
     +--ro lsp-log
     |  +--ro event* [id]
```

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

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

notifications:
  +---n database-overload
```

```
     |  +--ro routing-protocol-name?   -> /rt:routing/
     |  |                                 control-plane-protocols/
     |  |                                 control-plane-protocol/name
     |  +--ro isis-level?              level
     |  +--ro overload?                enumeration
     +---n lsp-too-large
     |  +--ro routing-protocol-name?   -> /rt:routing/
     |  |                                 control-plane-protocols/
     |  |                                 control-plane-protocol/name
     |  +--ro isis-level?              level
     |  +--ro interface-name?          if:interface-ref
     |  +--ro interface-level?         level
     |  +--ro extended-circuit-id?     extended-circuit-id
     |  +--ro pdu-size?                uint32
     |  +--ro lsp-id?                  lsp-id
     +---n if-state-change
     |  +--ro routing-protocol-name?   -> /rt:routing/
     |  |                                 control-plane-protocols/
     |  |                                 control-plane-protocol/name
     |  +--ro isis-level?              level
     |  +--ro interface-name?          if:interface-ref
     |  +--ro interface-level?         level
     |  +--ro extended-circuit-id?     extended-circuit-id
     |  +--ro state?                   if-state-type
     +---n corrupted-lsp-detected
     |  +--ro routing-protocol-name?   -> /rt:routing/
     |  |                                 control-plane-protocols/
     |  |                                 control-plane-protocol/name
     |  +--ro isis-level?              level
     |  +--ro lsp-id?                  lsp-id
     +---n attempt-to-exceed-max-sequence
     |  +--ro routing-protocol-name?   -> /rt:routing/
     |  |                                 control-plane-protocols/
     |  |                                 control-plane-protocol/name
     |  +--ro isis-level?              level
     |  +--ro lsp-id?                  lsp-id
     +---n id-len-mismatch
     |  +--ro routing-protocol-name?   -> /rt:routing/
     |  |                                 control-plane-protocols/
     |  |                                 control-plane-protocol/name
     |  +--ro isis-level?              level
     |  +--ro interface-name?          if:interface-ref
     |  +--ro interface-level?         level
     |  +--ro extended-circuit-id?     extended-circuit-id
     |  +--ro pdu-field-len?           uint8
     |  +--ro raw-pdu?                 binary
     +---n max-area-addresses-mismatch
     |  +--ro routing-protocol-name?   -> /rt:routing/
```

```
     |  |                              control-plane-protocols/
     |  |                              control-plane-protocol/name
     |  +--ro isis-level?            level
     |  +--ro interface-name?        if:interface-ref
     |  +--ro interface-level?       level
     |  +--ro extended-circuit-id?   extended-circuit-id
     |  +--ro max-area-addresses?    uint8
     |  +--ro raw-pdu?               binary
     +---n own-lsp-purge
     |  +--ro routing-protocol-name?   -> /rt:routing/
     |  |                              control-plane-protocols/
     |  |                              control-plane-protocol/name
     |  +--ro isis-level?            level
     |  +--ro interface-name?        if:interface-ref
     |  +--ro interface-level?       level
     |  +--ro extended-circuit-id?   extended-circuit-id
     |  +--ro lsp-id?                lsp-id
     +---n sequence-number-skipped
     |  +--ro routing-protocol-name?   -> /rt:routing/
     |  |                              control-plane-protocols/
     |  |                              control-plane-protocol/name
     |  +--ro isis-level?            level
     |  +--ro interface-name?        if:interface-ref
     |  +--ro interface-level?       level
     |  +--ro extended-circuit-id?   extended-circuit-id
     |  +--ro lsp-id?                lsp-id
     +---n authentication-type-failure
     |  +--ro routing-protocol-name?   -> /rt:routing/
     |  |                              control-plane-protocols/
     |  |                              control-plane-protocol/name
     |  +--ro isis-level?            level
     |  +--ro interface-name?        if:interface-ref
     |  +--ro interface-level?       level
     |  +--ro extended-circuit-id?   extended-circuit-id
     |  +--ro raw-pdu?               binary
     +---n authentication-failure
     |  +--ro routing-protocol-name?   -> /rt:routing/
     |  |                              control-plane-protocols/
     |  |                             control-plane-protocol/name
     |  +--ro isis-level?            level
     |  +--ro interface-name?        if:interface-ref
     |  +--ro interface-level?       level
     |  +--ro extended-circuit-id?   extended-circuit-id
     |  +--ro raw-pdu?               binary
     +---n version-skew
     |  +--ro routing-protocol-name?   -> /rt:routing/
     |  |                              control-plane-protocols/
     |  |                              control-plane-protocol/name
```

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

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

2.1. IS-IS Configuration

The IS-IS configuration is divided into:

- o Global parameters.

- o Per-interface configuration (see Section 2.4).

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

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

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

2.2. Multi-topology Parameters

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

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

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

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

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

2.3. Per-Level Parameters

Some parameters allow a per-level configuration. For such parameters, 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>
          </priority>
```

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

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

Example of augmentation:

```
augment "/rt:routing/" +
        "rt:control-plane-protocols/rt:control-plane-protocol"+
        "/isis:isis/isis:overload" {
        when "rt:type = 'isis:isis'" {
          description
           "This 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 ([RFC8343].

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

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

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

```
          |  |  +--rw (authentication-type)?
          |  |     +--:(key-chain) {key-chain}?
          |  |     |  +--rw key-chain?          key-chain:key-chain-ref
          |  |     +--:(password)
          |  |        +--rw key?                string
          |  |        +--rw crypto-algorithm?   identityref
          |  +--rw level-2
          |     +--rw (authentication-type)?
          |        +--:(key-chain) {key-chain}?
          |        |  +--rw key-chain?          key-chain:key-chain-ref
          |        +--:(password)
          |           +--rw key?                string
          |           +--rw crypto-algorithm?   identityref
          +--rw hello-interval
          |  +--rw value?     rt-types:timer-value-seconds16
          |  +--rw level-1
          |  |  +--rw value?   rt-types:timer-value-seconds16
          |  +--rw level-2
          |     +--rw value?   rt-types:timer-value-seconds16
          +--rw hello-multiplier
          |  +--rw value?     uint16
          |  +--rw level-1
          |  |  +--rw value?   uint16
          |  +--rw level-2
          |     +--rw value?   uint16
          +--rw priority
          |  +--rw value?     uint8
          |  +--rw level-1
          |  |  +--rw value?   uint8
          |  +--rw level-2
          |     +--rw value?   uint8
          +--rw metric
          |  +--rw value?     wide-metric
          |  +--rw level-1
          |  |  +--rw value?   wide-metric
          |  +--rw level-2
          |     +--rw value?   wide-metric
          +--rw bfd {bfd}?
          |  +--rw enable?                           boolean
          |  +--rw local-multiplier?                 multiplier
          |  +--rw (interval-config-type)?
          |     +--:(tx-rx-intervals)
          |     |  +--rw desired-min-tx-interval?    uint32
          |     |  +--rw required-min-rx-interval?   uint32
          |     +--:(single-interval) {single-minimum-interval}?
          |        +--rw min-interval?               uint32
          +--rw address-families {nlpid-control}?
          |  +--rw address-family-list* [address-family]
```

```
          |     +--rw address-family    iana-rt-types:address-family
          +--rw mpls
          |  +--rw ldp
          |     +--rw igp-sync?   boolean {ldp-igp-sync}?
          +--rw fast-reroute {fast-reroute}?
          |  +--rw lfa {lfa}?
          |     +--rw candidate-enable?   boolean
          |     +--rw enable?             boolean
          |     +--rw remote-lfa {remote-lfa}?
          |     |  +--rw enable?   boolean
          |     +--rw level-1
          |     |  +--rw candidate-enable?   boolean
          |     |  +--rw enable?             boolean
          |     |  +--rw remote-lfa {remote-lfa}?
          |     |     +--rw enable?   boolean
          |     +--rw level-2
          |        +--rw candidate-enable?   boolean
          |        +--rw enable?             boolean
          |        +--rw remote-lfa {remote-lfa}?
          |           +--rw enable?   boolean
          +--ro adjacencies
          |  +--ro adjacency* []
          |     +--ro neighbor-sys-type?             level
          |     +--ro neighbor-sysid?                system-id
          |     +--ro neighbor-extended-circuit-id?  extended-circuit-id
          |     +--ro neighbor-snpa?                 snpa
          |     +--ro usage?                         level
          |     +--ro hold-timer?                    rt-types:
          |     |                                  timer-value-seconds16
          |     +--ro neighbor-priority?             uint8
          |     +--ro lastuptime?                    yang:timestamp
          |     +--ro state?                         adj-state-type
          +--ro event-counters
          |  +--ro adjacency-changes?             uint32
          |  +--ro adjacency-number?              uint32
          |  +--ro init-fails?                    uint32
          |  +--ro adjacency-rejects?             uint32
          |  +--ro id-len-mismatch?               uint32
          |  +--ro max-area-addresses-mismatch?   uint32
          |  +--ro authentication-type-fails?     uint32
          |  +--ro authentication-fails?          uint32
          |  +--ro lan-dis-changes?               uint32
          +--ro packet-counters
          |  +--ro level* [level]
          |     +--ro level      level-number
          |     +--ro iih
          |     |  +--ro in?    uint32
          |     |  +--ro out?   uint32
```

```
          |      +--ro ish
          |      |  +--ro in?    uint32
          |      |  +--ro out?   uint32
          |      +--ro esh
          |      |  +--ro in?    uint32
          |      |  +--ro out?   uint32
          |      +--ro lsp
          |      |  +--ro in?    uint32
          |      |  +--ro out?   uint32
          |      +--ro psnp
          |      |  +--ro in?    uint32
          |      |  +--ro out?   uint32
          |      +--ro csnp
          |      |  +--ro in?    uint32
          |      |  +--ro out?   uint32
          |      +--ro unknown
          |         +--ro in?   uint32
          +--rw discontinuity-time?        yang:date-and-time
          +--rw topologies {multi-topology}?
             +--rw topology* [name]
                +--rw name      ->
                |              ../../../../../../../../rt:ribs/rib/name
                +--rw metric
                   +--rw value?     wide-metric
                   +--rw level-1
                   |  +--rw value?   wide-metric
                   +--rw level-2
                      +--rw value?   wide-metric

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

notifications:
  +---n database-overload
  |  +--ro routing-protocol-name?   -> /rt:routing/
  |  |                                    control-plane-protocols/
```

```
     |  |                                   control-plane-protocol/name
     |  +--ro isis-level?             level
     |  +--ro overload?               enumeration
     +---n lsp-too-large
     |  +--ro routing-protocol-name?   -> /rt:routing/
     |  |                                   control-plane-protocols/
     |  |                                   control-plane-protocol/name
     |  +--ro isis-level?             level
     |  +--ro interface-name?         if:interface-ref
     |  +--ro interface-level?        level
     |  +--ro extended-circuit-id?    extended-circuit-id
     |  +--ro pdu-size?               uint32
     |  +--ro lsp-id?                 lsp-id
     +---n if-state-change
     |  +--ro routing-protocol-name?   -> /rt:routing/
     |  |                                   control-plane-protocols/
     |  |                                   control-plane-protocol/name
     |  +--ro isis-level?             level
     |  +--ro interface-name?         if:interface-ref
     |  +--ro interface-level?        level
     |  +--ro extended-circuit-id?    extended-circuit-id
     |  +--ro state?                  if-state-type
     +---n corrupted-lsp-detected
     |  +--ro routing-protocol-name?   -> /rt:routing/
     |  |                                   control-plane-protocols/
     |  |                                   control-plane-protocol/name
     |  +--ro isis-level?             level
     |  +--ro lsp-id?                 lsp-id
     +---n attempt-to-exceed-max-sequence
     |  +--ro routing-protocol-name?   -> /rt:routing/
     |  |                                   control-plane-protocols/
     |  |                                   control-plane-protocol/name
     |  +--ro isis-level?             level
     |  +--ro lsp-id?                 lsp-id
     +---n id-len-mismatch
     |  +--ro routing-protocol-name?   -> /rt:routing/
     |  |                                   control-plane-protocols/
     |  |                                   control-plane-protocol/name
     |  +--ro isis-level?             level
     |  +--ro interface-name?         if:interface-ref
     |  +--ro interface-level?        level
     |  +--ro extended-circuit-id?    extended-circuit-id
     |  +--ro pdu-field-len?          uint8
     |  +--ro raw-pdu?                binary
     +---n max-area-addresses-mismatch
     |  +--ro routing-protocol-name?   -> /rt:routing/
     |  |                                   control-plane-protocols/
     |  |                                   control-plane-protocol/name
```

```
  |  +--ro isis-level?               level
  |  +--ro interface-name?           if:interface-ref
  |  +--ro interface-level?          level
  |  +--ro extended-circuit-id?      extended-circuit-id
  |  +--ro max-area-addresses?       uint8
  |  +--ro raw-pdu?                  binary
  +---n own-lsp-purge
  |  +--ro routing-protocol-name?    -> /rt:routing/
  |  |                                      control-plane-protocols/
  |  |                                      control-plane-protocol/name
  |  +--ro isis-level?               level
  |  +--ro interface-name?           if:interface-ref
  |  +--ro interface-level?          level
  |  +--ro extended-circuit-id?      extended-circuit-id
  |  +--ro lsp-id?                   lsp-id
  +---n sequence-number-skipped
  |  +--ro routing-protocol-name?    -> /rt:routing/
  |  |                                      control-plane-protocols/
  |  |                                      control-plane-protocol/name
  |  +--ro isis-level?               level
  |  +--ro interface-name?           if:interface-ref
  |  +--ro interface-level?          level
  |  +--ro extended-circuit-id?      extended-circuit-id
  |  +--ro lsp-id?                   lsp-id
  +---n authentication-type-failure
  |  +--ro routing-protocol-name?    -> /rt:routing/
  |  |                                      control-plane-protocols/
  |  |                                      control-plane-protocol/name
  |  +--ro isis-level?               level
  |  +--ro interface-name?           if:interface-ref
  |  +--ro interface-level?          level
  |  +--ro extended-circuit-id?      extended-circuit-id
  |  +--ro raw-pdu?                  binary
  +---n authentication-failure
  |  +--ro routing-protocol-name?    -> /rt:routing/
  |  |                                      control-plane-protocols/
  |  |                                      control-plane-protocol/name
  |  +--ro isis-level?               level
  |  +--ro interface-name?           if:interface-ref
  |  +--ro interface-level?          level
  |  +--ro extended-circuit-id?      extended-circuit-id
  |  +--ro raw-pdu?                  binary
  +---n version-skew
  |  +--ro routing-protocol-name?    -> /rt:routing/
  |  |                                      control-plane-protocols/
  |  |                                      control-plane-protocol/name
  |  +--ro isis-level?               level
  |  +--ro interface-name?           if:interface-ref
```

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

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

2.5. Authentication Parameters

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

2.6. IGP/LDP synchronization

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

2.7. ISO parameters

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

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

The clns-mtu can be configured for an interface.

2.8. IP FRR

This YANG module supports LFA (Loop Free Alternates) [RFC5286] and remote LFA [RFC7490] as IP Fast Re-Route (FRR) techniques. The "fast-reroute" container may be augmented by other models to support other IP FRR flavors (MRT as defined in [RFC7812], TI-LFA as defined in [I-D.ietf-rtgwg-segment-routing-ti-lfa], etc.).

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

Remote LFA is considered as an extension of LFA. Remote LFA cannot be enabled if LFA is not enabled.

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

2.9. Operational States

Operational state is defined in module in various containers at various levels:

- o system-counters: Provides statistical information about the global system.

- o interface: Provides configuration state information for each interface.

- o adjacencies: Provides state information about current IS-IS adjacencies.

- spf-log: Provides information about SPF events for an IS-IS instance. This SHOULD be implemented as a wrapping buffer.

- lsp-log: Provides information about LSP events for an IS-IS instance (reception of an LSP or modification of a local LSP). This SHOULD be implemented as a wrapping buffer and the implementation MAY optionally log LSP refreshes.

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

- database: Provides contents of the current Link State Database.

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

- fast-reroute: Provides IP FRR state information.

3. RPC Operations

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

- clear-database: Reset the content of a particular IS-IS database and restart database synchronization with all neighbors.

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

4. Notifications

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

database-overload: This notification is sent when the IS-IS Node overload condition changes.

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

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

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

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

id-len-mismatch: This notification is sent when 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 an LSP with a parse error.

adjacency-state-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 an LSP is received.

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

5. Interaction with Other YANG Modules

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

The "isis" module augments "/if:interfaces/if:interface" defined by [RFC8343] 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 with IS-IS-specific operational states.

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

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

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

To support BFD for fast detection, the module relies on [I-D.ietf-bfd-yang].

6. IS-IS YANG Module

The following RFCs, drafts and external standards are not referenced in the document text but are referenced in the ietf-isis.yang module: [ISO-10589], [RFC1195], [RFC4090],[RFC5029], [RFC5130], [RFC5302], [RFC5305], [RFC5306], [RFC5307], [RFC5308], [RFC5880], [RFC5881], [RFC6119], [RFC6232], [RFC7794], [RFC7981], [RFC8570], [RFC7917], [RFC8405].

```
<CODE BEGINS> file "ietf-isis@2019-10-15.yang"
module ietf-isis {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-isis";

  prefix isis;

  import ietf-routing {
    prefix "rt";
    reference "RFC 8349 - A YANG Data Model for Routing
               Management (NMDA Version)";
  }

  import ietf-inet-types {
    prefix inet;
    reference "RFC 6991 - Common YANG Data Types";
  }

  import ietf-yang-types {
```

```
       prefix yang;
       reference "RFC 6991 - Common YANG Data Types";
     }

     import ietf-interfaces {
       prefix "if";
       reference "RFC 8343 - A YANG Data Model for Interface
                  Management (NDMA Version)";
     }

     import ietf-key-chain {
       prefix "key-chain";
       reference "RFC 8177 - YANG Data Model for Key Chains";
     }

     import ietf-routing-types {
       prefix "rt-types";
       reference "RFC 8294 - Common YANG Data Types for the
                  Routing Area";
     }

     import iana-routing-types {
       prefix "iana-rt-types";
       reference "RFC 8294 - Common YANG Data Types for the
                  Routing Area";
     }

     import ietf-bfd-types {
       prefix "bfd-types";
       reference "RFC YYYY - YANG Data Model for Bidirectional
                  Forwarding Detection (BFD).

   -- Note to RFC Editor Please replace YYYY with published RFC
     number for draft-ietf-bfd-yang.";

     }

     organization
       "IETF LSR Working Group";

     contact
       "WG Web:   <https://datatracker.ietf.org/group/lsr/>
        WG List:  <mailto:lsr@ietf.org>

        Editor:   Stephane Litkowski
                  <mailto:slitkows.ietf@gmail.com>
        Author:   Derek Yeung
                  <mailto:derek@arrcus.com>
```

```
        Author:   Acee Lindem
                  <mailto:acee@cisco.com>
        Author:   Jeffrey Zhang
                  <mailto:zzhang@juniper.net>
        Author:   Ladislav Lhotka
                  <mailto:llhotka@nic.cz>";

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

       This YANG model conforms to the Network Management
       Datastore Architecture (NMDA) as described in RFC 8242.

       Copyright (c) 2018 IETF Trust and the persons identified as
       authors of the code.  All rights reserved.

       Redistribution and use in source and binary forms, with or
       without modification, is permitted pursuant to, and subject to
       the license terms contained in, the Simplified BSD License set
       forth in Section 4.c of the IETF Trust's Legal Provisions
       Relating to IETF Documents
       (https://trustee.ietf.org/license-info).

       This version of this YANG module is part of RFC XXXX
       (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
       for full legal notices.

       The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL
       NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'NOT RECOMMENDED',
       'MAY', and 'OPTIONAL' in this document are to be interpreted as
       described in BCP 14 (RFC 2119) (RFC 8174) when, and only when,
       they appear in all capitals, as shown here.

       This version of this YANG module is part of RFC XXXX;
       see the RFC itself for full legal notices.";

     revision 2019-10-15 {
       description
         "Initial revision.";
       reference "RFC XXXX";
     }

     /* Identities */
```

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

     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.";
     }

     identity frr-protection-method {
       description
         "Base identity for a Fast Reroute protection method.";
     }
     identity frr-protection-method-lfa {
       base frr-protection-method;
       description "Loop Free Alternate as defined in RFC5286.";
     }
     identity frr-protection-method-rlfa {
       base frr-protection-method;
       description "Remote Loop Free Alternate as defined in RFC7490.";
     }
     identity frr-protection-method-rsvpte {
       base frr-protection-method;
       description "RSVP-TE as defined in RFC4090.";
     }

     identity frr-protection-available-type {
       description "Base identity for Fast Reroute protection types
                    provided by an alternate path.";
     }
     identity frr-protection-available-node-type {
       base frr-protection-available-type;
       description "Node protection is provided by the alternate.";
     }
```

```
     identity frr-protection-available-link-type {
       base frr-protection-available-type;
       description "Link protection is provided by the alternate.";
     }
     identity frr-protection-available-srlg-type {
       base frr-protection-available-type;
       description "SRLG protection is provided by the alternate.";
     }
     identity frr-protection-available-downstream-type {
       base frr-protection-available-type;
       description "The alternate is downstream of node in the path.";
     }
     identity frr-protection-available-other-type {
       base frr-protection-available-type;
       description "The level of protection is unknown.";
     }

     identity frr-alternate-type {
       description "Base identity for IP Fast Reroute alternate type.";
     }
     identity frr-alternate-type-equal-cost {
       base frr-alternate-type;
           description "ECMP alternate.";
     }
     identity frr-alternate-type-lfa {
       base frr-alternate-type;
           description "LFA alternate.";
     }
     identity frr-alternate-type-remote-lfa {
       base frr-alternate-type;
           description "Remote LFA alternate.";
     }
     identity frr-alternate-type-tunnel {
       base frr-alternate-type;
           description "Tunnel based alternate (such as,
                    RSVP-TE or GRE).";
     }
     identity frr-alternate-mrt {
       base frr-alternate-type;
           description "MRT alternate.";
     }
     identity frr-alternate-tilfa {
       base frr-alternate-type;
           description "TILFA alternate.";
     }
     identity frr-alternate-other {
       base frr-alternate-type;
           description "Other alternate.";
```

```
     }


     identity unidirectional-link-delay-subtlv-flag {
         description "Base identity for unidirectional-link-delay
                     subTLV flags. Flags are defined in RFC8570.";
     }
     identity unidirectional-link-delay-subtlv-a-flag {
         base unidirectional-link-delay-subtlv-flag;
         description
             "The A bit represents the Anomalous (A) bit.
               The A bit is set when the measured value of
               this parameter exceeds its configured
               maximum threshold.
               The A bit is cleared when the measured value
               falls below its configured reuse threshold.
               If the A bit is clear,
               the value represents steady-state link performance.";
     }
     identity min-max-unidirectional-link-delay-subtlv-flag {
         description
           "Base identity for min-max-unidirectional-link-delay
            subTLV flags. Flags are defined in RFC8570.";
     }
     identity min-max-unidirectional-link-delay-subtlv-a-flag {
         base min-max-unidirectional-link-delay-subtlv-flag;
         description
             "The A bit represents the Anomalous (A) bit.
               The A bit is set when the measured value of
               this parameter exceeds its configured
               maximum threshold.
               The A bit is cleared when the measured value
               falls below its configured reuse threshold.
               If the A bit is clear,
               the value represents steady-state link performance.";
     }
     identity unidirectional-link-loss-subtlv-flag {
         description "Base identity for unidirectional-link-loss
                     subTLV flags. Flags are defined in RFC8570.";
     }

     identity unidirectional-link-loss-subtlv-a-flag {
         base unidirectional-link-loss-subtlv-flag;
         description
             "The A bit represents the Anomalous (A) bit.
               The A bit is set when the measured value of
               this parameter exceeds its configured
               maximum threshold.
```

```
               The A bit is cleared when the measured value
               falls below its configured reuse threshold.
               If the A bit is clear,
               the value represents steady-state link performance.";
     }
     identity tlv229-flag {
         description "Base identity for TLV229 flags. Flags are defined
                    in RFC5120.";
     }
     identity tlv229-overload-flag {
         base tlv229-flag;
         description
             "If set, the originator is overloaded,
             and must be avoided in path calculation.";
     }
     identity tlv229-attached-flag {
         base tlv229-flag;
         description
             "If set, the originator is attached to
             another area using the referred metric.";
     }
     identity router-capability-flag {
         description "Base identity for router capability flags.
           Flags are defined in RFC7981.";
     }
     identity router-capability-flooding-flag {
         base router-capability-flag;
         description
             "Quote from RFC7981: 'If the S bit is set,
              the IS-IS Router CAPABILITY
              TLV MUST be flooded across the entire routing
              domain. If the S bit is clear, the TLV MUST NOT
              be leaked between levels. This bit MUST NOT
              be altered during the TLV leaking'.";
     }
     identity router-capability-down-flag {
         base router-capability-flag;
         description
             "Quote from RFC7981: '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 in to prevent
              TLV looping'.";
     }

     identity lsp-flag {
         description "Base identity for LSP attributes.
```

```
                      Attributes are defined in ISO 10589";
      }
      identity lsp-partitioned-flag {
          base lsp-flag;
          description "Originator partition repair supported";
      }
      identity lsp-attached-error-metric-flag {
          base lsp-flag;
          description "Set when originator is attached to
               another area using the error metric.";
      }
      identity lsp-attached-delay-metric-flag {
          base lsp-flag;
          description "Set when originator is attached to
               another area using the delay metric.";
      }
      identity lsp-attached-expense-metric-flag {
          base lsp-flag;
          description "Set when originator is attached to
               another area using the expense metric.";
      }
      identity lsp-attached-default-metric-flag {
          base lsp-flag;
          description "Set when originator is attached to
               another area using the default metric.";
      }
      identity lsp-overload-flag {
          base lsp-flag;
          description
              "If set, the originator is overloaded,
               and must be avoided in path calculation.";
      }
      identity lsp-l1system-flag {
          base lsp-flag;
          description
              "Set when the Intermediate System has an L1 type.";
      }
      identity lsp-l2system-flag {
          base lsp-flag;
          description
              "Set when the Intermediate System has an L2 type.";
      }


      /* Feature definitions */

      feature osi-interface {
        description "Support of OSI specific parameters on an
```

```
                   interface.";
     }
     feature poi-tlv {
       description "Support of Purge Originator Identification.";
       reference "RFC 6232 - Purge Originator Identification TLV
                  for IS-IS";
     }
     feature ietf-spf-delay {
       description
         "Support for IETF SPF delay algorithm.";
       reference "RFC 8405 - SPF Back-off algorithm for link
                  state IGPs";
     }
     feature bfd {
       description
         "Support for BFD detection of IS-IS neighbor reachability.";
       reference "RFC 5880 - Bidirectional Forwarding Detection (BFD)
                  RFC 5881 - Bidirectional Forwarding Detection
                  (BFD) for IPv4 and IPv6 (Single Hop)";
     }
     feature key-chain {
       description
         "Support of keychain for authentication.";
       reference "RFC8177 - YANG Data Model for Key Chains";
     }
     feature node-flag {
       description
         "Support for node-flag for IS-IS prefixes.";
       reference "RFC7794 - IS-IS Prefix Attributes for
                  Extended IP and IPv6 Reachability";
     }
     feature node-tag {
       description
         "Support for node admin tag for IS-IS routing instances.";
       reference "RFC7917 - Advertising Node Administrative Tags
                  in IS-IS";
     }
     feature ldp-igp-sync {
       description
         "Support for LDP IGP synchronization.";
       reference "RFC5443 - LDP IGP Synchronization.";
     }
     feature fast-reroute {
       description
         "Support for IP Fast Reroute (IP-FRR).";
     }
     feature nsr {
       description
```

```
         "Support for Non-Stop-Routing (NSR). The IS-IS NSR feature
             allows  a router with redundant control-plane capability
             (e.g., dual Route-Processor (RP) cards) to maintain its
             state and adjacencies during planned and unplanned
             IS-IS instance restarts. It differs from graceful-restart
             or Non-Stop Forwarding (NSF) in that no protocol signaling
             or assistance from adjacent IS-IS neighbors is required to
             recover control-plane state.";
     }
     feature lfa {
       description
         "Support for Loop-Free Alternates (LFAs).";
       reference "RFC5286 - Basic Specification of IP Fast-Reroute:
                  Loop-free Alternates";
     }
     feature remote-lfa {
       description
         "Support for Remote Loop-Free Alternates (R-LFAs).";
       reference "RFC7490 - Remote Loop-Free Alternate Fast Reroute";
     }

     feature overload-max-metric {
       description
         "Support of overload by setting all links to max metric.
           In IS-IS, the overload bit is usually used to signal that
           a node cannot be used as a transit. The overload-max-metric
           feature brings a similar behavior leveraging on setting all
           the link metrics to MAX_METRIC.";
     }
     feature prefix-tag {
       description
         "Support for 32-bit prefix tags";
       reference "RFC5130 - A Policy Control Mechanism in
                  IS-IS Using Administrative Tags";
     }
     feature prefix-tag64 {
       description
         "Support for 64-bit prefix tags";
       reference "RFC5130 - A Policy Control Mechanism in
                  IS-IS Using Administrative Tags";
     }
     feature auto-cost {
       description
         "Support for IS-IS interface metric computation
          according to a reference bandwidth.";
     }

     feature te-rid {
```

```
       description
         "Traffic-Engineering Router-ID.";
       reference "RFC5305 - IS-IS Extensions for Traffic Engineering
                  RFC6119 - IPv6 Traffic Engineering in IS-IS";
     }
     feature max-ecmp {
       description
         "Setting maximum number of ECMP paths.";
     }
     feature multi-topology {
       description
         "Support for Multiple-Topology Routing (MTR).";
       reference "RFC5120 - M-IS-IS: Multi Topology Routing in IS-IS";
     }
     feature nlpid-control {
       description
         "Support for the advertisement
          of a Network Layer Protocol Identifier within IS-IS
          configuration.";
     }
     feature graceful-restart {
       description
         "IS-IS Graceful restart support.";
       reference "RFC5306 - Restart Signaling in IS-IS";
     }

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

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

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

     /* Type definitions */

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

```
     }

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

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

     }

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

     }
```

```
     typedef adj-state-type {
       type enumeration {
         enum "up" {
           description
             "State indicates the adjacency is established.";
         }
         enum "down" {
           description
             "State indicates the adjacency is NOT established.";
         }
         enum "init" {
           description
             "State indicates the adjacency is establishing.";
         }
         enum "failed" {
           description
             "State indicates the 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 the state of an interface";
     }

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

     typedef lsp-id {
       type string {
         pattern
```

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

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

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

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

```
          system-id.
          An example system-id is 0143.0438.AEF0.00";
     }

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

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

     typedef mesh-group-state {
       type enumeration {
         enum "mesh-inactive" {
           description
             "Interface is not part of a mesh group.";
         }
         enum "mesh-set" {
           description
             "Interface is part of a mesh group.";
         }
         enum "mesh-blocked" {
           description
             "LSPs must not be flooded over this interface.";
         }
       }
       description
         "This type describes mesh group state of an interface";
     }

     /* Grouping for notifications */

     grouping notification-instance-hdr {
       description
         "Instance specific IS-IS notification data grouping";
       leaf routing-protocol-name {
         type leafref {
           path "/rt:routing/rt:control-plane-protocols/"
              + "rt:control-plane-protocol/rt:name";
```

```
         }
         description "Name of the IS-IS instance.";
       }
       leaf isis-level {
         type level;
         description "IS-IS level of the instance.";
       }
     }

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


    /* Groupings for IP Fast Reroute */

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

```
    grouping interface-lfa-config {
       leaf candidate-enable {
         type boolean;
         default "true";
         description
           "Enable the interface to be used as backup.";
       }
       leaf enable {
         type boolean;
         default false;
         description
           "Activates LFA - Per-prefix LFA computation
            is assumed.";
       }
       container remote-lfa {
         if-feature remote-lfa;
         leaf enable {
           type boolean;
           default false;
           description
             "Activates Remote LFA (R-LFA).";
         }
         description
           "Remote LFA configuration.";
       }
       description "Grouping for LFA interface configuration";
     }
     grouping interface-fast-reroute-config {
       description
         "This group defines interface configuration of IP-FRR.";
       container fast-reroute {
         if-feature fast-reroute;
         container lfa {
           if-feature lfa;
           uses interface-lfa-config;
           container level-1 {
             uses interface-lfa-config;
             description
               "LFA level 1 config";
           }
           container level-2 {
           uses interface-lfa-config;
            description
             "LFA level 2 config";
           }
           description
             "LFA configuration.";
         }
```

```
         description
           "Interface IP Fast-reroute configuration.";
       }
     }
     grouping instance-fast-reroute-state {
       description "IPFRR state data grouping";
       container protected-routes {
         config false;
         list address-family-stats {
           key "address-family prefix alternate";

           leaf address-family {
             type iana-rt-types:address-family;
             description
               "Address-family";
           }
           leaf prefix {
             type inet:ip-prefix;
             description
               "Protected prefix.";
           }
           leaf alternate {
             type inet:ip-address;
             description
               "Alternate next hop for the prefix.";
           }
           leaf alternate-type {
                   type identityref {
                     base frr-alternate-type;
                   }
             description
               "Type of alternate.";
           }
           leaf best {
             type boolean;
             description
               "Is set when the alternate is the preferred one,
                is clear otherwise.";
           }
           leaf non-best-reason {
             type string {
               length "1..255";
             }
             description
               "Information field to describe why the alternate
                is not best. The length should be limited to 255
                unicode characters. The expected format is a single
                line text.";
```

```
          }
          container protection-available {
            leaf-list protection-types {
              type identityref {
                base frr-protection-available-type;
              }
              description "This list contains a set of protection
                          types defined as identities.
                          An identity must be added for each type of
                          protection provided by the alternate.
                          As an example, if an alternate provides
                          SRLG, node and link protection, three
                          identities must be added in this list:
                          one for SRLG protection, one for node
                          protection, one for link protection.";
            }
            description "Protection types provided by the alternate.";
          }
          leaf alternate-metric1 {
            type uint32;
            description
              "Metric from Point of Local Repair (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 protected prefix statistics.";
        }
        description
          "List of prefixes that are protected.";
      }

      container unprotected-routes {
        config false;
        list prefixes {
          key "address-family prefix";

          leaf address-family {
            type iana-rt-types:address-family;
```

```
            description "Address-family";
          }
          leaf prefix {
            type inet:ip-prefix;
            description "Unprotected prefix.";
          }
          description
            "Per-AF unprotected prefix statistics.";
        }
        description
          "List of prefixes that are not protected.";
      }

      list protection-statistics {
        key frr-protection-method;
        config false;
        leaf frr-protection-method {
          type identityref {
            base frr-protection-method;
          }
          description "Protection method used.";
        }
        list address-family-stats {
          key address-family;

          leaf address-family {
            type iana-rt-types:address-family;

            description "Address-family";
          }
          leaf total-routes {
            type yang:gauge32;
            description "Total prefixes.";
          }
          leaf unprotected-routes {
            type yang:gauge32;
            description
              "Total prefixes that are not protected.";
          }
          leaf protected-routes {
            type yang:gauge32;
            description
              "Total prefixes that are protected.";
          }
          leaf link-protected-routes {
            type yang:gauge32;
            description
              "Total prefixes that are link protected.";
```

```
          }
          leaf node-protected-routes {
            type yang:gauge32;
            description
              "Total prefixes that are node protected.";
          }
          description
            "Per-AF protected prefix statistics.";
        }

        description "Global protection statistics.";
      }
    }

          /* Route table and local RIB groupings */

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

```
           }
           leaf level {
             type level-number;
             description "Level number for this route.";
           }
           leaf route-tag {
             type uint32;
             description "Route tag for this route.";
           }
         }
       }
     }

     grouping route-content {
       description
         "IS-IS protocol-specific route properties grouping.";
       leaf metric {
         type uint32;
         description "IS-IS metric of a route.";
       }
       leaf-list tag {
         type uint64;
         description
           "List of tags associated with the route.
            This list provides a consolidated view of both
            32-bit and 64-bit tags (RFC5130) available for the prefix.";
       }
       leaf route-type {
         type enumeration {
           enum l2-intra-area {
             description "Level 2 internal route. As per RFC5302,
                          the prefix is directly connected to the
                          advertising router. It cannot be
                          distinguished from an L1->L2 inter-area
                          route.";
           }
           enum l1-intra-area {
             description "Level 1 internal route. As per RFC5302,
                          the prefix is directly connected to the
                          advertising router.";
           }
           enum l2-external {
             description "Level 2 external route. As per RFC5302,
                          such a route is learned from other IGPs.
                          It cannot be distinguished from an L1->L2
                          inter-area external route.";
           }
           enum l1-external {
```

```
            description "Level 1 external route. As per RFC5302,
                         such a route is learned from other IGPs.";
          }
          enum l1-inter-area {
            description "These prefixes are learned via L2 routing.";
          }
          enum l1-inter-area-external {
            description "These prefixes are learned via L2 routing
                         towards an l2-external route.";
          }
        }
        description "IS-IS route type.";
      }
    }


          /* Grouping definitions for configuration and ops state */


    grouping adjacency-state {
      container adjacencies {
        config false;
        list adjacency {
          leaf neighbor-sys-type {
            type level;
            description
              "Level capability 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
              "Define the level(s) activated for the adjacency.
               On a p2p link this might be level 1 and 2,
```

```
                but on a LAN, the usage will be level 1
                between neighbors at level 1 or level 2 between
                neighbors at level 2.";
           }
           leaf hold-timer {
             type rt-types:timer-value-seconds16;
             units seconds;
             description
               "The holding time in seconds for this
                adjacency. This value is based on
                received hello PDUs and the elapsed
                time since receipt.";
           }
           leaf neighbor-priority {
             type uint8 {
               range "0 .. 127";
             }
             description
               "Priority of the neighboring IS for becoming
                the 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 admin-control {
       leaf enable {
         if-feature admin-control;
         type boolean;
         default "true";
         description
           "Enable/Disable the protocol.";
       }
       description
         "Grouping for admin control.";
     }

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

       leaf hold-down {
         type rt-types:timer-value-milliseconds;
         units msec;
         description
           "Timer used to consider an IGP stability period
                                (milliseconds).";
       }
       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 (milliseconds).";
       }
       leaf current-state {
         type enumeration {
```

```
           enum "quiet" {
             description "QUIET state";
           }
           enum "short-wait" {
             description "SHORT_WAIT state";
           }
           enum "long-wait" {
             description "LONG_WAIT state";
           }
         }
         config false;
         description
           "Current SPF back-off algorithm state.";
       }
       leaf remaining-time-to-learn {
         type rt-types:timer-value-milliseconds;
         units "msec";
         config false;
         description
           "Remaining time until time-to-learn timer fires.";
       }
       leaf remaining-hold-down {
         type rt-types:timer-value-milliseconds;
         units "msec";
         config false;
         description
           "Remaining time until hold-down timer fires.";
       }
       leaf last-event-received {
         type yang:timestamp;
         config false;
         description
           "Time of last IGP event received";
       }
       leaf next-spf-time {
         type yang:timestamp;
         config false;
         description
           "Time when next SPF has been scheduled.";
       }
       leaf last-spf-time {
         type yang:timestamp;
         config false;
         description
           "Time of last SPF computation.";
       }
       description
         "Grouping for IETF SPF delay configuration and state.";
```

```
     }



     grouping node-tag-config {
       description
         "IS-IS node tag config state.";
       container node-tags {
         if-feature node-tag;
         list node-tag {
           key tag;
           leaf tag {
             type uint32;
               description
                 "Node tag value.";
           }
           description
             "List of tags.";
         }
         description
           "Container for node admin tags.";
       }
     }

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

```
           }
         }
         description "Choice of authentication.";
       }
       description "Grouping for global authentication 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
           "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,
            - both means that both are advertised.
            This leaf is only affecting IPv4 metrics.";
       }
       description
         "Grouping for global metric style config.";
     }

     grouping metric-type-global-cfg-with-default {
       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";
```

```
           }
         }
         default wide-only;
         description
           "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,
            - 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;
         description  "Value of the metric";
       }
       description
         "Global default metric config grouping.";
     }
     grouping default-metric-global-cfg-with-default {
       leaf value {
         type wide-metric;
         default "10";
         description  "Value of the metric";
       }
       description
         "Global default metric config grouping.";
     }


     grouping overload-global-cfg {
       leaf status {
         type boolean;
         default false;
         description
           "This leaf specifies 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
           "Timeout (in seconds) of the overload condition.";
       }
       description
         "Overload maximum metric configuration grouping";
     }

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

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

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

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

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

      description "Interval between hello messages.";
    }

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

```
       description
           "Number of missed hello messages prior to
            adjacency down grouping.";
     }
     grouping hello-multiplier-cfg-with-default {
       leaf value {
         type uint16;
         default 3;
         description
           "Number of missed hello messages prior to
            declaring the adjacency down.";
       }
       description
           "Number of missed hello messages prior to
            adjacency down grouping.";
     }

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

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

       description "Interface DIS election priority grouping";
     }

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

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


     grouping metric-parameters {
       container metric-type {
         uses metric-type-global-cfg-with-default;
         container level-1 {
           uses metric-type-global-cfg;
           description "level-1 specific configuration";
         }
         container level-2 {
           uses metric-type-global-cfg;
           description "level-2 specific configuration";
         }
         description "Metric style global configuration";
       }

       container default-metric {
         uses default-metric-global-cfg-with-default;
         container level-1 {
           uses default-metric-global-cfg;
           description "level-1 specific configuration";
         }
         container level-2 {
           uses default-metric-global-cfg;
           description "level-2 specific configuration";
         }
         description "Default metric global configuration";
       }
       container auto-cost {
         if-feature auto-cost;
         description
           "Interface Auto-cost configuration state.";
         leaf enable {
           type boolean;
           description
             "Enable/Disable interface auto-cost.";
         }
         leaf reference-bandwidth {
           when "../enable = 'true'" {
             description "Only when auto cost is enabled";
```

```
          }
          type uint32 {
            range "1..4294967";
          }
          units Mbits;
          description
            "Configure reference bandwidth used to automatically
             determine interface cost (Mbits). The cost is the
             reference bandwidth divided by the interface speed
             with 1 being the minimum cost.";
        }
      }

      description "Grouping for global metric parameters.";
    }

    grouping high-availability-parameters {
      container graceful-restart {
        if-feature graceful-restart;
        leaf enable {
          type boolean;
          default false;
          description "Enable graceful restart.";
        }
        leaf restart-interval {
          type rt-types:timer-value-seconds16;
          units "seconds";
          description
            "Interval (in seconds) to attempt graceful restart prior
             to failure.";
        }
        leaf helper-enable {
          type boolean;
          default "true";
          description
            "Enable local IS-IS router as graceful restart helper.";
        }
        description "Graceful-Restart Configuration.";
      }
      container nsr {
        if-feature nsr;
        description "Non-Stop Routing (NSR) configuration.";
        leaf enable {
          type boolean;
          default false;
          description "Enable/Disable Non-Stop Routing (NSR).";
        }
      }
```

```
       description "Grouping for High Availability parameters.";
     }

     grouping authentication-parameters {
       container authentication {
         uses authentication-global-cfg;

         container level-1 {
           uses authentication-global-cfg;
           description "level-1 specific configuration";
         }
         container level-2 {
           uses authentication-global-cfg;
           description "level-2 specific configuration";
         }
         description "Authentication global configuration for
                      both LSPs and SNPs.";
       }
       description "Grouping for authentication parameters";
     }
     grouping address-family-parameters {
       container address-families {
         if-feature nlpid-control;
         list address-family-list {
           key address-family;
           leaf address-family {
             type iana-rt-types:address-family;
             description "Address-family";
           }
           leaf enable {
             type boolean;
             description "Activate the address family.";
           }
           description
             "List of address families and whether or not they
              are activated.";
         }
         description "Address Family configuration";
       }
       description "Grouping for address family parameters.";
     }

     grouping mpls-parameters {
       container mpls {
         container te-rid {
           if-feature te-rid;
           description
             "Stable ISIS Router IP Address used for Traffic
```

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

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

```
       }
       leaf poi-tlv {
         if-feature poi-tlv;
         type boolean;
         default false;
         description
           "Enable advertisement of IS-IS Purge Originator
                   Identification TLV.";
       }
       description "Grouping for LSP global parameters.";
     }
     grouping spf-parameters {
       container spf-control {
           leaf paths {
             if-feature max-ecmp;
             type uint16 {
               range "1..65535";
             }
             description
               "Maximum number of Equal-Cost Multi-Path (ECMP) paths.";
           }
           container ietf-spf-delay {
             if-feature ietf-spf-delay;
             uses ietf-spf-delay;
             description "IETF SPF delay algorithm configuration.";
           }
           description
             "SPF calculation control.";
       }
       description "Grouping for SPF global parameters.";
     }
     grouping instance-config {
       description "IS-IS global configuration grouping";

       uses admin-control;

       leaf level-type {
         type level;
         default "level-all";
         description
           "Level of an IS-IS node - can be level-1,
            level-2 or level-all.";
       }

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

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

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

       uses lsp-parameters;
       uses high-availability-parameters;
       uses node-tag-config;
       uses metric-parameters;
       uses authentication-parameters;
       uses address-family-parameters;
       uses mpls-parameters;
       uses spf-parameters;
       uses instance-fast-reroute-config;

       container preference {
         uses route-preference-global-cfg;
         description "Router preference configuration for IS-IS
                      protocol instance route installation";
       }

       container overload {
         uses overload-global-cfg;
         description "Router protocol instance overload state
                      configuration";
       }

       container overload-max-metric {
         if-feature overload-max-metric;
         uses overload-max-metric-global-cfg;
         description
           "Router protocol instance overload maximum
            metric advertisement configuration.";
       }
     }

     grouping instance-state {
       description
         "IS-IS instance operational state.";
       uses spf-log;
```

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

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

      grouping interface-config {
        description "Interface configuration grouping";

        uses admin-control;

        leaf level-type {
          type level;
          default "level-all";
          description "IS-IS level of the interface.";
        }
        leaf lsp-pacing-interval {
          type rt-types:timer-value-milliseconds;
```

```
         units "milliseconds";
         default 33;
         description
           "Interval (in milli-seconds) between LSP
            transmissions.";
       }
       leaf lsp-retransmit-interval {
         type rt-types:timer-value-seconds16;
         units "seconds";
         description
           "Interval (in seconds) between LSP
            retransmissions.";
       }
       leaf passive {
         type boolean;
         default "false";
         description
           "Indicates whether the interface is in passive mode (IS-IS
            not running but network is advertised).";
       }
       leaf csnp-interval {
         type rt-types:timer-value-seconds16;
         units "seconds";
         default 10;
         description
            "Interval (in seconds) between CSNP messages.";
       }
       container hello-padding {
         leaf enable {
           type boolean;
           default "true";
           description
             "IS-IS Hello-padding activation - enabled by default.";
         }
         description "IS-IS hello padding configuration.";
       }
       leaf mesh-group-enable {
         type mesh-group-state;
         description "IS-IS interface mesh-group state";
       }
       leaf mesh-group {
         when "../mesh-group-enable = 'mesh-set'" {
           description
             "Only valid when mesh-group-enable equals mesh-set";
         }
         type uint8;
         description "IS-IS interface mesh-group ID.";
       }
```

```
       leaf interface-type {
         type interface-type;
         default "broadcast";
         description
           "Type of adjacency to be established for the interface. This
            dictates the type of hello messages that are used.";
       }

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

```
          description "level-2 specific configuration";
        }
        description "Interval between hello messages.";
      }
      container hello-multiplier {
        uses hello-multiplier-cfg-with-default;
        container level-1 {
          uses hello-multiplier-cfg;
          description "level-1 specific configuration";
        }
        container level-2 {
          uses hello-multiplier-cfg;
          description "level-2 specific configuration";
        }
        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-with-default;
        container level-1 {
          uses priority-cfg;
          description "level-1 specific configuration";
        }
        container level-2 {
          uses priority-cfg;
          description "level-2 specific configuration";
        }
        description "Priority for DIS election.";
      }
      container metric {
        uses metric-cfg-with-default;
        container level-1 {
          uses metric-cfg;
          description "level-1 specific configuration";
        }
        container level-2 {
          uses metric-cfg;
          description "level-2 specific configuration";
        }
        description "Metric configuration.";
      }
      container bfd {
        if-feature bfd;
        description "BFD Client Configuration.";
```

```
         uses bfd-types:client-cfg-parms;

         reference "RFC YYYY - YANG Data Model for Bidirectional
                  Forwarding Detection (BFD).

   -- Note to RFC Editor Please replace YYYY with published FC
      number for draft-ietf-bfd-yang.";

       }
       container address-families {
         if-feature nlpid-control;
         list address-family-list {
           key address-family;
           leaf address-family {
             type iana-rt-types:address-family;
             description  "Address-family";
           }
           description "List of AFs.";
         }
         description "Interface address-families";
       }
       container mpls {
         container ldp {
           leaf igp-sync {
             if-feature ldp-igp-sync;
             type boolean;
             default false;
             description "Enables IGP/LDP synchronization";
           }
           description "LDP protocol related configuration.";
         }
         description "MPLS configuration for IS-IS interfaces";
       }
       uses interface-fast-reroute-config;
     }

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

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

    /* Grouping for the hostname database */

     grouping hostname-db {
       container hostnames {
         config false;
         list hostname {
           key system-id;
           leaf system-id {
             type system-id;
             description
               "system-id associated with the hostname.";
           }
           leaf hostname {
             type string {
               length "1..255";
             }
             description
               "Hostname associated with the system-id
                as defined in RFC5301.";
           }
           description
             "List of system-id/hostname associations.";
         }
         description
           "Hostname to system-id mapping database.";
       }
```

```
       description
         "Grouping for hostname to system-id mapping database.";
     }

     /* Groupings for counters */

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

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

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

```
     grouping event-counters {
       container event-counters {
         config false;
         leaf adjacency-changes {
           type uint32;
           description
             "The number of times an adjacency state change has
              occurred on this interface.";
         }
         leaf adjacency-number {
           type uint32;
           description
             "The number of adjacencies on this interface.";
         }
         leaf init-fails {
           type uint32;
           description
             "The number of times initialization of this
              interface has failed. This counts events such
              as 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 has been
              received on this interface with the
              max area address field differing from that of
              this system.";
         }
         leaf authentication-type-fails {
           type uint32;
           description
             "Number of authentication type mismatches.";
         }
```

```
          leaf authentication-fails {
            type uint32;
            description
              "Number of authentication key failures.";
          }
          leaf lan-dis-changes {
            type uint32;
            description
              "The number of times the DIS has changed on this
               interface at this level. If the interface type is
               point-to-point, the count is zero.";
          }
          description "IS-IS interface event counters.";
        }
        description
          "Grouping for IS-IS interface event counters";
      }

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

            leaf level {
              type level-number;
              description "IS-IS level.";
            }
            container iih {
              leaf in {
                type uint32;
                description "Received IIH PDUs.";
              }
              leaf out {
                type uint32;
                description "Sent IIH PDUs.";
              }
              description "Number of IIH PDUs received/sent.";
            }
            container ish {
              leaf in {
                type uint32;
                description "Received ISH PDUs.";
              }
              leaf out {
                type uint32;
                description "Sent ISH PDUs.";
              }
```

```
              description
                "ISH PDUs received/sent.";
            }
            container esh {
              leaf in {
                type uint32;
                description "Received ESH PDUs.";
              }
              leaf out {
                type uint32;
                description "Sent ESH PDUs.";
              }
              description "Number of ESH PDUs received/sent.";
            }
            container lsp {
              leaf in {
                type uint32;
                description "Received LSP PDUs.";
              }
              leaf out {
                type uint32;
                description "Sent LSP PDUs.";
              }
              description "Number of LSP PDUs received/sent.";
            }
            container psnp {
              leaf in {
                type uint32;
                description "Received PSNP PDUs.";
              }
              leaf out {
                type uint32;
                description "Sent PSNP PDUs.";
              }
              description "Number of PSNP PDUs received/sent.";
            }
            container csnp {
              leaf in {
                type uint32;
                description "Received CSNP PDUs.";
              }
              leaf out {
                type uint32;
                description "Sent CSNP PDUs.";
              }
              description "Number of CSNP PDUs received/sent.";
            }
            container unknown {
```

```
            leaf in {
              type uint32;
              description "Received unknown PDUs.";
            }
            description "Number of unknown PDUs received/sent.";
          }
          description
            "List of packet counter for supported levels.";
        }
        description "Packet counters per IS-IS level.";
      }
      description
        "Grouping for per IS-IS Level packet counters.";
    }

          /* Groupings for various log buffers */
    grouping spf-log {
      container spf-log {
        config false;
        list event {
          key id;

          leaf id {
            type yang:counter32;
            description
              "Event identifier -  purely internal value.
               It is expected the most recent events to have the bigger
               id number.";
          }
          leaf spf-type {
            type enumeration {
              enum full {
                description "Full SPF computation.";
              }
              enum route-only {
                description
                  "Route reachability only SPF computation";
              }
            }
            description "Type of SPF computation performed.";
          }
          leaf level {
            type level-number;
            description
              "IS-IS level number for SPF computation";
          }
          leaf schedule-timestamp {
            type yang:timestamp;
```

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

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

    grouping lsp-log {
      container lsp-log {
        config false;
        list event {
          key id;
```

```
           leaf id {
             type yang:counter32;
             description
               "Event identifier -  purely internal value.
                It is expected the most recent events to have the bigger
                id number.";
           }
           leaf level {
             type level-number;
             description
               "IS-IS level number for LSP";
           }
           container lsp {
             leaf lsp {
               type lsp-id;
               description
                 "LSP ID of the LSP.";
             }
             leaf sequence {
               type uint32;
               description
                 "Sequence number of the LSP.";
             }
             description
               "LSP identification container - either the received
                LSP or the locally generated LSP.";
           }

           leaf received-timestamp {
             type yang:timestamp;
             description
               "This is the timestamp when the LSA was received.
                In case of local LSA update, the timestamp refers
                to the LSA origination time.";
           }

           leaf reason {
             type identityref {
               base lsp-log-reason;
             }
             description "Type of LSP change.";
           }

           description
             "List of LSP events - implemented as a
              wrapping buffer.";
         }
```

```
        description
          "This container lists the LSP log.
           Local LSP modifications are also included
           in the list.";

    } description "Grouping for LSP log.";
  }



  /* Groupings for the LSDB description */

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

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

  /* TLVs and sub-TLVs for prefixes */

  grouping prefix-reachability-attributes {
    description
      "Grouping for extended reachability attributes of an
```

```
            IPv4 or IPv6 prefix.";

        leaf external-prefix-flag {
          type boolean;
          description "External prefix flag.";
        }
        leaf readvertisement-flag {
          type boolean;
          description "Re-advertisement flag.";
        }
        leaf node-flag {
          type boolean;
          description "Node flag.";
        }
      }

      grouping prefix-ipv4-source-router-id {
        description
          "Grouping for the IPv4 source router ID of a prefix
           advertisement.";

        leaf ipv4-source-router-id {
          type inet:ipv4-address;
          description "IPv4 Source router ID address.";
        }
      }

      grouping prefix-ipv6-source-router-id {
        description
          "Grouping for the IPv6 source router ID of a prefix
           advertisement.";

        leaf ipv6-source-router-id {
          type inet:ipv6-address;
          description "IPv6 Source router ID address.";
        }
      }

      grouping prefix-attributes-extension {
        description "Prefix extended attributes
                     as defined in RFC7794.";

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

      grouping prefix-ipv4-std {
```

```
      description
        "Grouping for attributes of an IPv4 standard prefix
         as defined in RFC1195.";
      leaf ip-prefix {
        type inet:ipv4-address;
        description "IPv4 prefix address";
      }
      leaf prefix-len {
        type uint8;
        description "IPv4 prefix length (in bits)";
      }
      leaf i-e {
        type boolean;
        description
          "Internal or External (I/E) Metric bit value.
           Set to 'false' to indicate an internal metric.";
      }
      container default-metric {
        leaf metric {
          type std-metric;
          description "Default IS-IS metric for IPv4 prefix";
        }
        description "IS-IS default metric container.";
      }
      container delay-metric {
        leaf metric {
          type std-metric;
          description "IS-IS delay metric for IPv4 prefix";
        }
        leaf supported {
          type boolean;
          default "false";
          description
            "Indicates whether IS-IS delay metric is supported.";
        }
        description "IS-IS delay metric container.";
      }
      container expense-metric {
        leaf metric {
          type std-metric;
          description "IS-IS expense metric for IPv4 prefix";
        }
        leaf supported {
          type boolean;
          default "false";
          description
            "Indicates whether IS-IS expense metric is supported.";
        }
```

```
          description "IS-IS expense metric container.";
        }
        container error-metric {
          leaf metric {
            type std-metric;
            description
              "This leaf describes the IS-IS error metric value";
          }
          leaf supported {
            type boolean;
            default "false";
            description
              "Indicates whether IS-IS error metric is supported.";
          }
          description "IS-IS error metric container.";
        }
      }

      grouping prefix-ipv4-extended {
        description
          "Grouping for attributes of an IPv4 extended prefix
           as defined in RFC5305.";
        leaf up-down {
          type boolean;
          description  "Value of up/down bit.
              Set to true when the prefix has been advertised down
              the hierarchy.";
        }
        leaf ip-prefix {
          type inet:ipv4-address;
          description "IPv4 prefix address";
        }
        leaf prefix-len {
          type uint8;
          description "IPv4 prefix length (in bits)";
        }
        leaf metric {
          type wide-metric;
          description "IS-IS wide metric value";
        }
        leaf-list tag {
          type uint32;
          description
            "List of 32-bit tags associated with the IPv4 prefix.";
        }
        leaf-list tag64 {
          type uint64;
          description
```

```
            "List of 64-bit tags associated with the IPv4 prefix.";
        }
        uses prefix-attributes-extension;
      }

      grouping prefix-ipv6-extended {
        description "Grouping for attributes of an IPv6 prefix
                     as defined in RFC5308.";
        leaf up-down {
          type boolean;
          description "Value of up/down bit.
              Set to true when the prefix has been advertised down
              the hierarchy.";
        }
        leaf ip-prefix {
          type inet:ipv6-address;
          description "IPv6 prefix address";
        }
        leaf prefix-len {
          type uint8;
          description  "IPv6 prefix length (in bits)";
        }
        leaf metric {
          type wide-metric;
          description  "IS-IS wide metric value";
        }
        leaf-list tag {
          type uint32;
          description
            "List of 32-bit tags associated with the IPv4 prefix.";
        }
        leaf-list tag64 {
          type uint64;
          description
            "List of 64-bit tags associated with the IPv4 prefix.";
        }
        uses prefix-attributes-extension;
      }

      /* TLVs and sub-TLVs for neighbors */

      grouping neighbor-link-attributes {
        description
          "Grouping for link attributes as defined
          in RFC5029";
        leaf link-attributes-flags {
          type uint16;
          description
```

```
            "Flags for the link attributes";
        }
      }
      grouping neighbor-gmpls-extensions {
        description
          "Grouping for GMPLS attributes of a neighbor as defined
          in RFC5307";
        leaf link-local-id {
          type uint32;
          description
            "Local identifier of the link.";
        }
        leaf remote-local-id {
          type uint32;
          description
            "Remote identifier of the link.";
        }
        leaf protection-capability {
          type uint8;
          description
            "Describes the protection capabilities
            of the link. This is the value of the
            first octet of the sub-TLV type 20 value.";
        }
        container interface-switching-capability {
          description
            "Interface switching capabilities of the link.";
          leaf switching-capability {
            type uint8;
            description
              "Switching capability of the link.";
          }
          leaf encoding {
            type uint8;
            description
              "Type of encoding of the LSP being used.";
          }
          container max-lsp-bandwidths {
            description "Per-priority max LSP bandwidths.";
            list max-lsp-bandwidth {
              leaf priority {
                type uint8 {
                  range "0 .. 7";
                }
                description "Priority from 0 to 7.";
              }
              leaf bandwidth {
                type rt-types:bandwidth-ieee-float32;
```

```
                description "max LSP bandwidth.";
              }
              description
                "List of max LSP bandwidths for different
                 priorities.";
            }
          }
          container tdm-specific {
            when "../switching-capability = 100";
            description
              "Switching Capability-specific information applicable
              when switching type is TDM.";

            leaf minimum-lsp-bandwidth {
              type rt-types:bandwidth-ieee-float32;
              description "minimum LSP bandwidth.";
            }
            leaf indication {
              type uint8;
              description
                "The indication whether the interface supports Standard
                or Arbitrary SONET/SDH.";
            }
          }
          container psc-specific {
            when "../switching-capability >= 1 and
                  ../switching-capability <= 4";
            description
              "Switching Capability-specific information applicable
              when switching type is PSC1,PSC2,PSC3 or PSC4.";

            leaf minimum-lsp-bandwidth {
              type rt-types:bandwidth-ieee-float32;
              description "minimum LSP bandwidth.";
            }
            leaf mtu {
              type uint16;
              units bytes;
              description
                "Interface MTU";
            }
          }
        }
      }

      grouping neighbor-extended-te-extensions {
        description
          "Grouping for TE attributes of a neighbor as defined
```

```
         in RFC8570";

       container unidirectional-link-delay {
         description
           "Container for the average delay
           from the local neighbor to the remote one.";
         container flags {
           leaf-list unidirectional-link-delay-subtlv-flags {
               type identityref {
                   base unidirectional-link-delay-subtlv-flag;
               }
              description
                    "This list contains identities for the bits
                     which are set.";
           }
           description
             "unidirectional-link-delay subTLV flags.";
         }
         leaf value {
           type uint32;
           units usec;
           description
             "Delay value expressed in microseconds.";
         }
       }
       container min-max-unidirectional-link-delay {
         description
           "Container for the min and max delay
           from the local neighbor to the remote one.";
         container flags {
           leaf-list min-max-unidirectional-link-delay-subtlv-flags {
               type identityref {
                   base min-max-unidirectional-link-delay-subtlv-flag;
               }
               description
                   "This list contains identities for the bits which are
                   set.";
           }
           description
             "min-max-unidirectional-link-delay subTLV flags.";
         }
         leaf min-value {
           type uint32;
           units usec;
           description
             "Minimum delay value expressed in microseconds.";
         }
         leaf max-value {
```

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

```
          description
            "Residual bandwidth.";
        }
      }
      container unidirectional-link-available-bandwidth {
        description
          "Container for the available bandwidth
          from the local neighbor to the remote one.";
        leaf value {
          type rt-types:bandwidth-ieee-float32;
          units Bps;
          description
            "Available bandwidth.";
        }
      }
      container unidirectional-link-utilized-bandwidth {
        description
          "Container for the utilized bandwidth
          from the local neighbor to the remote one.";
        leaf value {
          type rt-types:bandwidth-ieee-float32;
          units Bps;
          description
            "Utilized bandwidth.";
        }
      }
    }

    grouping neighbor-te-extensions {
      description
        "Grouping for TE attributes of a neighbor as defined
        in RFC5305";
      leaf admin-group {
        type uint32;
        description
          "Administrative group/Resource Class/Color.";
      }
      container local-if-ipv4-addrs {
        description "All local interface IPv4 addresses.";
        leaf-list local-if-ipv4-addr {
          type inet:ipv4-address;
          description
            "List of local interface IPv4 addresses.";
        }
      }
      container remote-if-ipv4-addrs {
        description "All remote interface IPv4 addresses.";
        leaf-list remote-if-ipv4-addr {
```

```
           type inet:ipv4-address;
           description
             "List of remote interface IPv4 addresses.";
         }
       }
       leaf te-metric {
         type uint32;
         description "TE metric.";
       }
       leaf max-bandwidth {
         type rt-types:bandwidth-ieee-float32;
         description "Maximum bandwidth.";
       }
       leaf max-reservable-bandwidth {
         type rt-types:bandwidth-ieee-float32;
         description "Maximum reservable bandwidth.";
       }
       container unreserved-bandwidths {
         description "All unreserved bandwidths.";
         list unreserved-bandwidth {
           leaf priority {
             type uint8 {
               range "0 .. 7";
             }
             description "Priority from 0 to 7.";
           }
           leaf unreserved-bandwidth {
             type rt-types:bandwidth-ieee-float32;
             description "Unreserved bandwidth.";
           }
           description
             "List of unreserved bandwidths for different
              priorities.";
         }
       }
     }

     grouping neighbor-extended {
       description
        "Grouping for attributes of an IS-IS extended neighbor.";
       leaf neighbor-id {
         type extended-system-id;
         description "system-id of the extended neighbor.";
       }
      container instances {
         description "List of all adjacencies between the local
                      system and the neighbor system-id.";
         list instance {
```

```
          key id;

          leaf id {
            type uint32;
            description "Unique identifier of an instance of a
                         particular neighbor.";
          }
          leaf metric {
            type wide-metric;
            description "IS-IS wide metric for extended neighbor";
          }
          uses neighbor-gmpls-extensions;
          uses neighbor-te-extensions;
          uses neighbor-extended-te-extensions;
          uses neighbor-link-attributes;
          uses unknown-tlvs;
          description "Instance of a particular adjacency.";
        }
      }
    }

    grouping neighbor {
      description  "IS-IS standard neighbor grouping.";
      leaf neighbor-id {
        type extended-system-id;
        description "IS-IS neighbor system-id";
      }
     container instances {
        description "List of all adjacencies between the local
                     system and the neighbor system-id.";
        list instance {
          key id;

          leaf id {
            type uint32;
            description "Unique identifier of an instance of a
                         particular neighbor.";
          }
          leaf i-e {
            type boolean;
            description
              "Internal or External (I/E) Metric bit value.
               Set to 'false' to indicate an internal metric.";
          }
          container default-metric {
            leaf metric {
              type std-metric;
              description "IS-IS default metric value";
```

```
            }
           description "IS-IS default metric container";
          }
          container delay-metric {
            leaf metric {
              type std-metric;
              description "IS-IS delay metric value";
            }
            leaf supported {
              type boolean;
              default "false";
              description "IS-IS delay metric supported";
            }
            description "IS-IS delay metric container";
          }
          container expense-metric {
            leaf metric {
              type std-metric;
              description "IS-IS expense metric value";
            }
            leaf supported {
              type boolean;
              default "false";
              description "IS-IS expense metric supported";
            }
            description "IS-IS expense metric container";
          }
          container error-metric {
            leaf metric {
              type std-metric;
              description "IS-IS error metric value";
            }
            leaf supported {
              type boolean;
              default "false";
              description "IS-IS error metric supported";
            }
            description "IS-IS error metric container";
          }
          description "Instance of a particular adjacency
                       as defined in ISO10589.";
        }
      }
    }

    /* Top-level TLVs */

    grouping tlv132-ipv4-addresses {
```

```
       leaf-list ipv4-addresses {
         type inet:ipv4-address;
         description
           "List of IPv4 addresses of the IS-IS node - IS-IS
            reference is TLV 132.";
       }
       description "Grouping for TLV132.";
     }
     grouping tlv232-ipv6-addresses {
       leaf-list ipv6-addresses {
         type inet:ipv6-address;
         description
           "List of IPv6 addresses of the IS-IS node - IS-IS
            reference is TLV 232.";
       }
       description "Grouping for TLV232.";
     }
     grouping tlv134-ipv4-te-rid {
       leaf ipv4-te-routerid {
         type inet:ipv4-address;
         description
           "IPv4 Traffic Engineering router ID of the IS-IS node -
            IS-IS reference is TLV 134.";
       }
       description "Grouping for TLV134.";
     }
     grouping tlv140-ipv6-te-rid {
       leaf ipv6-te-routerid {
         type inet:ipv6-address;
         description
           "IPv6 Traffic Engineering router ID of the IS-IS node -
            IS-IS reference is TLV 140.";
       }
       description "Grouping for TLV140.";
     }
     grouping tlv129-protocols {
       leaf-list protocol-supported {
         type uint8;
         description
         "List of supported protocols of the IS-IS node -
          IS-IS reference is TLV 129.";
       }
       description "Grouping for TLV129.";
     }
     grouping tlv137-hostname {
       leaf dynamic-hostname {
         type string;
         description
```

```
          "Host Name of the IS-IS node - IS-IS reference
           is TLV 137.";
      }
      description "Grouping for TLV137.";
    }
    grouping tlv10-authentication {
      container authentication {
        leaf authentication-type {
          type identityref {
              base key-chain:crypto-algorithm;
          }
          description
            "Authentication type to be used with IS-IS node.";
        }
        leaf authentication-key {
          type string;
          description
            "Authentication key to be used. For security reasons,
             the authentication key MUST NOT be presented in
             a clear text format in response to any request
            (e.g., via get, get-config).";
        }
        description
          "IS-IS node authentication information container -
           IS-IS reference is TLV 10.";
      }
      description "Grouping for TLV10.";
    }
    grouping tlv229-mt {
      container mt-entries {
        list topology {
          description
            "List of topologies supported";

          leaf mt-id {
            type uint16 {
              range "0 .. 4095";
            }
            description
              "Multi-Topology identifier of topology.";
          }
          container attributes {
              leaf-list flags {
                  type identityref {
                      base tlv229-flag;
                  }
                  description
                  "This list contains identities for the bits which are
```

```
                   set.";
               }
               description
                   "TLV 229 flags.";
           }
         }
         description
           "IS-IS node topology information container -
            IS-IS reference is TLV 229.";
       }
       description "Grouping for TLV229.";
     }

     grouping tlv242-router-capabilities {
       container router-capabilities {
         list router-capability {
             container flags {
               leaf-list router-capability-flags {
                   type identityref {
                       base router-capability-flag;
                   }
                   description
                   "This list contains identities for the bits which are
                   set.";
               }
               description
                   "Router capability flags.";
             }
             container node-tags {
               if-feature node-tag;
               list node-tag {
                 leaf tag {
                   type uint32;
                   description "Node tag value.";
                 }
                 description "List of tags.";
               }
               description "Container for node admin tags";
             }

             uses unknown-tlvs;

             description
               "IS-IS node capabilities. This list element may
                be extended with detailed information -  IS-IS
                reference is TLV 242.";
           }
           description "List of router capability TLVs.";
```

```
         }
         description "Grouping for TLV242.";
     }

     grouping tlv138-srlg {
       description
         "Grouping for TLV138.";
       container links-srlgs {
         list links {
           leaf neighbor-id {
             type extended-system-id;
             description "system-id of the extended neighbor.";
           }
           leaf flags {
             type uint8;
             description
               "Flags associated with the link.";
           }
           leaf link-local-id {
             type union {
               type inet:ip-address;
               type uint32;
             }
             description
               "Local identifier of the link.
               It could be an IPv4 address or a local identifier.";
           }
           leaf link-remote-id {
             type union {
               type inet:ip-address;
               type uint32;
             }
             description
               "Remote identifier of the link.
               It could be an IPv4 address or a remotely learned
               identifier.";
           }
           container srlgs {
             description "List of SRLGs.";
             leaf-list srlg {
               type uint32;
               description
                 "SRLG value of the link.";
             }
           }
           description
             "SRLG attribute of a link.";
         }
```

```
         description
           "List of links with SRLGs";
       }
     }

     /* Grouping for LSDB description */

     grouping lsp-entry {
       description "IS-IS LSP database entry grouping";

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

```
           description "LSP attributes.";
         }

         uses tlv132-ipv4-addresses;
         uses tlv232-ipv6-addresses;
         uses tlv134-ipv4-te-rid;
         uses tlv140-ipv6-te-rid;
         uses tlv129-protocols;
         uses tlv137-hostname;
         uses tlv10-authentication;
         uses tlv229-mt;
         uses tlv242-router-capabilities;
         uses tlv138-srlg;
         uses unknown-tlvs;

         container is-neighbor {
           list neighbor {
             key neighbor-id;

             uses neighbor;
             description "List of neighbors.";
           }
           description
             "Standard IS neighbors container - IS-IS reference is
              TLV 2.";
         }

         container extended-is-neighbor {
           list neighbor {
             key neighbor-id;

             uses neighbor-extended;
             description
               "List of extended IS neighbors";
           }
           description
             "Standard IS extended neighbors container - IS-IS
              reference is TLV 22";
         }

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

```
       }

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

       container extended-ipv4-reachability {
         list prefixes {
           uses prefix-ipv4-extended;
           uses unknown-tlvs;
           description "List of prefixes.";
         }
         description
           "IPv4 extended reachability information container -
            IS-IS reference is TLV 135.";
       }

       container mt-is-neighbor {
         list neighbor {
           leaf mt-id {
             type uint16 {
               range "0 .. 4095";
             }
             description "Multi-topology (MT) identifier";
           }
           uses neighbor-extended;
           description "List of neighbors.";
         }
         description
           "IS-IS multi-topology neighbor container - IS-IS
            reference is TLV 223.";
       }

       container mt-extended-ipv4-reachability {
         list prefixes {
           leaf mt-id {
             type uint16 {
               range "0 .. 4095";
             }
             description  "Multi-topology (MT) identifier";
           }
           uses prefix-ipv4-extended;
```

```
           uses unknown-tlvs;
           description "List of extended prefixes.";
         }
         description
           "IPv4 multi-topology (MT) extended reachability
            information container - IS-IS reference is TLV 235.";
       }

       container mt-ipv6-reachability {
         list prefixes {
           leaf MT-ID {
             type uint16 {
               range "0 .. 4095";
             }
             description "Multi-topology (MT) identifier";
           }
           uses prefix-ipv6-extended;
           uses unknown-tlvs;
           description "List of IPv6 extended prefixes.";
         }
         description
           "IPv6 multi-topology (MT) extended reachability
            information container - IS-IS reference is TLV 237.";
       }

       container ipv6-reachability {
         list prefixes {
           uses prefix-ipv6-extended;
           uses unknown-tlvs;
           description "List of IPv6 prefixes.";
         }
         description
           "IPv6 reachability information container - IS-IS
            reference is TLV 236.";
       }
     }

     grouping lsdb {
       description "Link State Database (LSDB) grouping";
       container database {
         config false;
         list levels {
           key level;

           leaf level {
             type level-number;
             description "LSDB level number (1 or 2)";
           }
```

```
          list lsp {
            key lsp-id;
            uses lsp-entry;
            description "List of LSPs in LSDB";
          }
          description "List of LSPs for the LSDB level container";
        }
        description "IS-IS Link State database container";
      }
    }



    /* Augmentations */

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


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

    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 IS-IS
         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 instance-config;
        uses instance-state;

        container topologies {
          if-feature multi-topology;
          list topology {
            key "name";
            leaf enable {
              type boolean;
              description "Topology enable configuration";
            }
            leaf name {
              type leafref {
                path "../../../../../../rt:ribs/rt:rib/rt:name";
              }
              description
                "Routing Information Base (RIB) corresponding
                 to topology.";
            }

            uses multi-topology-config;

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

              description
                "Reference to the interface within
                 the routing-instance.";
            }
            uses interface-config;
            uses interface-state;
            container topologies {
              if-feature multi-topology;
              list topology {
```

```
                 key name;

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

                   description
                     "Routing Information Base (RIB) corresponding
                      to topology.";
                 }
                 uses multi-topology-interface-config;
                 description "List of interface topologies";
               }
               description "Multi-topology container";
             }
             description "List of IS-IS interfaces.";
           }
           description
             "IS-IS interface specific configuration container";
         }

         description
           "IS-IS configuration/state top-level container";
       }
     }


     /* RPC methods */

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

         leaf routing-protocol-instance-name {
           type leafref {
             path "/rt:routing/rt:control-plane-protocols/"
                + "rt:control-plane-protocol/rt:name";
           }
           mandatory "true";
           description
             "Name of the IS-IS protocol instance whose IS-IS
              adjacency is being cleared.
```

```
              If the corresponding IS-IS instance doesn't exist,
              then the operation will fail with an error-tag of
              'data-missing' and an error-app-tag of
              'routing-protocol-instance-not-found'.";
         }
         leaf level {
           type level;
           description
             "IS-IS level of the adjacency to be cleared. If the
              IS-IS level is level-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 the operation
              SHALL fail with an error-tag of 'data-missing' and
              an error-app-tag of 'bad-isis-level'.";
         }
         leaf interface {
           type if:interface-ref;
           description
             "IS-IS interface name.

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

     rpc clear-database {
       description
         "This RPC request clears a particular  IS-IS database. If
          the operation fails for an IS-IS internal reason, then
          the error-tag and error-app-tag should be set
          indicating the reason for the failure.";
       input {
         leaf routing-protocol-instance-name {
           type leafref {
             path "/rt:routing/rt:control-plane-protocols/"
                + "rt:control-plane-protocol/rt:name";
           }
           mandatory "true";
           description
             "Name of the IS-IS protocol instance whose IS-IS
              database(s) is/are being cleared.

              If the corresponding IS-IS instance doesn't exist,
```

```
              then the operation will fail with an error-tag of
              'data-missing' and an error-app-tag of
              'routing-protocol-instance-not-found'.";
         }
         leaf level {
           type level;
           description
             "IS-IS level of the adjacency to be cleared. If the
              IS-IS level is level-1-2, both level 1 and level 2
              databases would be cleared.

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


     /* Notifications */

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

       leaf overload {
         type enumeration {
           enum off {
             description
               "Indicates IS-IS instance has left overload state";
           }
           enum on {
             description
               "Indicates IS-IS instance has entered overload state";
           }

         }
         description "New overload state of the IS-IS instance";
       }
       description
         "This notification is sent when an IS-IS instance
          overload state changes.";
     }

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

```
        leaf pdu-size {
          type uint32;
          description "Size of the LSP PDU";
        }
        leaf lsp-id {
          type lsp-id;
          description "LSP ID";
        }
        description
          "This notification is sent when we attempt to propagate
           an LSP that is larger than the dataLinkBlockSize (ISO10589)
           for the circuit.  The notification generation must be
           throttled with at least 5 seconds between successive
           notifications.";
      }

      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 5 seconds between successive
          notifications.";
     }

     notification max-area-addresses-mismatch {
       uses notification-instance-hdr;
       uses notification-interface-hdr;

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

     notification own-lsp-purge {
       uses notification-instance-hdr;
       uses notification-interface-hdr;
       leaf lsp-id {
```

```
         type lsp-id;
         description "LSP ID";
       }
       description
         "This notification is sent when the system receives
          a PDU with its own system-id and zero age.";
     }

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

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

     notification authentication-failure {
       uses notification-instance-hdr;
       uses notification-interface-hdr;
       leaf raw-pdu {
         type binary;
         description "Received raw PDU.";
       }
       description
         "This notification is sent when the system receives
          a PDU with the wrong authentication information.
          The notification generation must be throttled
```

```
          with at least 5 seconds between successive
          notifications.";
     }

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

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

     notification rejected-adjacency {
       uses notification-instance-hdr;
       uses notification-interface-hdr;
       leaf raw-pdu {
         type binary;
         description
           "Received raw PDU.";
       }
       leaf reason {
         type string {
```

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


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

    notification lsp-error-detected {
      uses notification-instance-hdr;
      uses notification-interface-hdr;
      leaf lsp-id {
        type lsp-id;
        description "LSP ID.";
      }
      leaf raw-pdu {
        type binary;
        description "Received raw PDU.";
      }
```

```
       leaf error-offset {
         type uint32;
         description
           "If the problem is a malformed TLV, the error-offset
            points to the start of the TLV. If the problem is with
            the LSP header, the error-offset points to the errant
            byte";
       }
       leaf tlv-type {
         type uint8;
         description
           "If the problem is a malformed TLV, the tlv-type is set
            to the type value of the suspicious TLV. Otherwise,
            this leaf is not present.";
       }
       description
         "This notification is sent when the system receives an
          LSP with a parse error. The notification generation must
          be throttled with at least 5 seconds between successive
          notifications.";
     }

     notification adjacency-state-change {
       uses notification-instance-hdr;
       uses notification-interface-hdr;
       leaf neighbor {
         type string {
           length "1..255";
         }
         description
           "Name of the neighbor.
            It corresponds to the hostname associated
            with the system-id of the neighbor in the
            mapping database (RFC5301).
            If the name of the neighbor is
            not available, it is not returned.";
       }
       leaf neighbor-system-id {
         type system-id;
         description "Neighbor system-id";
       }
       leaf state {
         type adj-state-type;

         description "New state of the IS-IS adjacency.";
       }
       leaf reason {
         type string {
```

```
            length "1..255";
          }
          description
            "If the adjacency is going to 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. The expected format is a single line
             text.";
        }
        description
          "This notification is sent when an IS-IS 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 "Timestamp when the LSP was received.";
        }
        leaf neighbor-system-id {
          type system-id;
          description "Neighbor system-id of LSP sender";
        }
        description
          "This notification is sent when an LSP is received.
           The notification generation must be throttled with at
           least 5 seconds between successive notifications.";
       }

      notification lsp-generation {
        uses notification-instance-hdr;

        leaf lsp-id {
          type lsp-id;
          description "LSP ID";
        }
```

```
        leaf sequence {
          type uint32;
          description "Sequence number of the received LSP.";
        }
        leaf send-timestamp {
          type yang:timestamp;

          description "Timestamp when our LSP was regenerated.";
        }
        description
          "This notification is sent when an LSP is regenerated.
           The notification generation must be throttled with at
           least 5 seconds between successive notifications.";
      }
    }
    <CODE ENDS>
```

7. Security Considerations

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

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

There are a number of data nodes defined in ietf-isis.yang module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. Writable data node represent configuration of each instance and interface. These correspond to the following schema nodes:

/isis

/isis/interfaces/interface[name]

For IS-IS, the ability to modify IS-IS configuration will allow the entire IS-IS domain to be compromised including forming adjacencies with unauthorized routers to misroute traffic or mount a massive

Denial-of-Service (DoS) attack. For example, adding IS-IS on any unprotected interface could allow an IS-IS adjacency to be formed with an unauthorized and malicious neighbor. Once an adjacency is formed, traffic could be hijacked. As a simpler example, a Denial-Of-Service attack could be mounted by changing the cost of an IS-IS interface to be asymmetric such that a hard routing loop ensues. In general, unauthorized modification of most IS-IS features will pose their own set of security risks and the "Security Considerations" in the respective reference RFCs should be consulted.

Some of the readable data nodes in the ietf-isis.yang module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. The exposure of the Link State Database (LSDB) will expose the detailed topology of the network. Similarly, the IS-IS local RIB exposes the reachable prefixes in the IS-IS routing domain. The Link State Database (LSDB) and local RIB are represented by the following schema nodes:

/isis/database

/isis/local-rib

Exposure of the Link State Database and local RIB include information beyond the scope of the IS-IS router and this may be undesirable since exposure may facilitate other attacks. Additionally, the complete IP network topology and, if deployed, the traffic engineering topology of the IS-IS domain can be reconstructed from the Link State Database. Though not as straightforward, the IS-IS local RIB can also be discover topological information. Network operators may consider their topologies to be sensitive confidential data.

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

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

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

The model describes several notifications, implementations must rate-limit the generation of these notifications to avoid creating significant notification load. Otherwise, this notification load may have some side effects on the system stability and may be exploited as an attack vector.

8. Contributors

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

9. Acknowledgements

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

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: The IESG
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. References

11.1. Normative References

[I-D.ietf-bfd-yang]
Rahman, R., Zheng, L., Jethanandani, M., Networks, J., and G. Mirsky, "YANG Data Model for Bidirectional Forwarding Detection (BFD)", draft-ietf-bfd-yang-17 (work in progress), August 2018.

[ISO-10589]
"Intermediate System to Intermediate System intra- domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service (ISO 8473)", International Standard 10589: 2002, Second Edition, 2002.

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

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

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

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

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

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

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

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

[RFC5301] McPherson, D. and N. Shen, "Dynamic Hostname Exchange Mechanism for IS-IS", RFC 5301, DOI 10.17487/RFC5301, October 2008, <https://www.rfc-editor.org/info/rfc5301>.

[RFC5302] Li, T., Smit, H., and T. Przygienda, "Domain-Wide Prefix Distribution with Two-Level IS-IS", RFC 5302, DOI 10.17487/RFC5302, October 2008, <https://www.rfc-editor.org/info/rfc5302>.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <https://www.rfc-editor.org/info/rfc5305>.

[RFC5306] Shand, M. and L. Ginsberg, "Restart Signaling for IS-IS", RFC 5306, DOI 10.17487/RFC5306, October 2008, <https://www.rfc-editor.org/info/rfc5306>.

[RFC5307] Kompella, K., Ed. and Y. Rekhter, Ed., "IS-IS Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 5307, DOI 10.17487/RFC5307, October 2008, <https://www.rfc-editor.org/info/rfc5307>.

[RFC5308] Hopps, C., "Routing IPv6 with IS-IS", RFC 5308, DOI 10.17487/RFC5308, October 2008, <https://www.rfc-editor.org/info/rfc5308>.

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

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <https://www.rfc-editor.org/info/rfc5880>.

[RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881, DOI 10.17487/RFC5881, June 2010, <https://www.rfc-editor.org/info/rfc5881>.

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

[RFC6119] Harrison, J., Berger, J., and M. Bartlett, "IPv6 Traffic Engineering in IS-IS", RFC 6119, DOI 10.17487/RFC6119, February 2011, <https://www.rfc-editor.org/info/rfc6119>.

[RFC6232] Wei, F., Qin, Y., Li, Z., Li, T., and J. Dong, "Purge Originator Identification TLV for IS-IS", RFC 6232, DOI 10.17487/RFC6232, May 2011, <https://www.rfc-editor.org/info/rfc6232>.

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

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

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

[RFC7490] Bryant, S., Filsfils, C., Previdi, S., Shand, M., and N. So, "Remote Loop-Free Alternate (LFA) Fast Reroute (FRR)", RFC 7490, DOI 10.17487/RFC7490, April 2015, <https://www.rfc-editor.org/info/rfc7490>.

[RFC7794] Ginsberg, L., Ed., Decraene, B., Previdi, S., Xu, X., and U. Chunduri, "IS-IS Prefix Attributes for Extended IPv4 and IPv6 Reachability", RFC 7794, DOI 10.17487/RFC7794, March 2016, <https://www.rfc-editor.org/info/rfc7794>.

[RFC7917] Sarkar, P., Ed., Gredler, H., Hegde, S., Litkowski, S., and B. Decraene, "Advertising Node Administrative Tags in IS-IS", RFC 7917, DOI 10.17487/RFC7917, July 2016, <https://www.rfc-editor.org/info/rfc7917>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[RFC7981] Ginsberg, L., Previdi, S., and M. Chen, "IS-IS Extensions for Advertising Router Information", RFC 7981, DOI 10.17487/RFC7981, October 2016, <https://www.rfc-editor.org/info/rfc7981>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8177] Lindem, A., Ed., Qu, Y., Yeung, D., Chen, I., and J. Zhang, "YANG Data Model for Key Chains", RFC 8177, DOI 10.17487/RFC8177, June 2017, <https://www.rfc-editor.org/info/rfc8177>.

[RFC8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Common YANG Data Types for the Routing Area", RFC 8294, DOI 10.17487/RFC8294, December 2017, <https://www.rfc-editor.org/info/rfc8294>.

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <https://www.rfc-editor.org/info/rfc8341>.

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <https://www.rfc-editor.org/info/rfc8342>.

[RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <https://www.rfc-editor.org/info/rfc8343>.

[RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <https://www.rfc-editor.org/info/rfc8349>.

[RFC8405] Decraene, B., Litkowski, S., Gredler, H., Lindem, A., Francois, P., and C. Bowers, "Shortest Path First (SPF) Back-Off Delay Algorithm for Link-State IGPs", RFC 8405, DOI 10.17487/RFC8405, June 2018, <https://www.rfc-editor.org/info/rfc8405>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8570] Ginsberg, L., Ed., Previdi, S., Ed., Giacalone, S., Ward, D., Drake, J., and Q. Wu, "IS-IS Traffic Engineering (TE) Metric Extensions", RFC 8570, DOI 10.17487/RFC8570, March 2019, <https://www.rfc-editor.org/info/rfc8570>.

11.2. Informative References

[I-D.ietf-rtgwg-segment-routing-ti-lfa] Litkowski, S., Bashandy, A., Filsfils, C., Decraene, B., Francois, P., daniel.voyer@bell.ca, d., Clad, F., and P. Camarillo, "Topology Independent Fast Reroute using Segment Routing", draft-ietf-rtgwg-segment-routing-ti-lfa-01 (work in progress), March 2019.

[RFC7812] Atlas, A., Bowers, C., and G. Enyedi, "An Architecture for IP/LDP Fast Reroute Using Maximally Redundant Trees (MRT-FRR)", RFC 7812, DOI 10.17487/RFC7812, June 2016, <https://www.rfc-editor.org/info/rfc7812>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <https://www.rfc-editor.org/info/rfc8340>.

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>192.0.2.1</router-id>
      <control-plane-protocols>
        <control-plane-protocol>
          <name>ISIS-example</name>
          <description/>
          <type>
            <type xmlns:isis="urn:ietf:params:xml:ns:yang:ietf-isis">
            isis:isis
            </type>
          </type>
          <isis xmlns="urn:ietf:params:xml:ns:yang:ietf-isis">
              <enable>true</enable>
              <level-type>level-2</level-type>
              <system-id>87FC.FCDF.4432</system-id>
              <area-address>49.0001</area-address>
              <mpls>
```

```
                   &lt;te-rid&gt;
                     &lt;ipv4-router-id&gt;192.0.2.1&lt;/ipv4-router-id&gt;
                   &lt;/te-rid&gt;
                 &lt;/mpls&gt;
                 &lt;lsp-lifetime&gt;65535&lt;/lsp-lifetime&gt;
                 &lt;lsp-refresh&gt;65000&lt;/lsp-refresh&gt;
                 &lt;metric-type&gt;
                   &lt;value&gt;wide-only&lt;/value&gt;
                 &lt;/metric-type&gt;
                 &lt;default-metric&gt;
                   &lt;value&gt;111111&lt;/value&gt;
                 &lt;/default-metric&gt;
                 &lt;address-families&gt;
                   &lt;address-family-list&gt;
                     &lt;address-family&gt;ipv4&lt;/address-family&gt;
                     &lt;enable&gt;true&lt;/enable&gt;
                   &lt;/address-family-list&gt;
                   &lt;address-family-list&gt;
                     &lt;address-family&gt;ipv6&lt;/address-family&gt;
                     &lt;enable&gt;true&lt;/enable&gt;
                   &lt;/address-family-list&gt;
                 &lt;/address-families&gt;
                 &lt;interfaces&gt;
                   &lt;interface&gt;
                     &lt;name&gt;Loopback0&lt;/name&gt;
                     &lt;tag&gt;200&lt;/tag&gt;
                     &lt;metric&gt;
                       &lt;value&gt;0&lt;/value&gt;
                     &lt;/metric&gt;
                     &lt;passive&gt;true&lt;/passive&gt;
                   &lt;/interface&gt;
                   &lt;interface&gt;
                     &lt;name&gt;Eth1&lt;/name&gt;
                     &lt;level-type&gt;level-2&lt;/level-type&gt;
                     &lt;interface-type&gt;point-to-point&lt;/interface-type&gt;
                     &lt;metric&gt;
                       &lt;value&gt;167890&lt;/value&gt;
                     &lt;/metric&gt;
                   &lt;/interface&gt;
                 &lt;/interfaces&gt;
             &lt;/isis&gt;
           &lt;/control-plane-protocol&gt;
         &lt;/control-plane-protocols&gt;
     &lt;/routing&gt;
     &lt;interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"&gt;
       &lt;interface&gt;
         &lt;name&gt;Loopback0&lt;/name&gt;
         &lt;description/&gt;
```

```
         <type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type">
         ianaift:softwareLoopback
         </type>
         <link-up-down-trap-enable>enabled</link-up-down-trap-enable>
         <ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
           <address>
             <ip>192.0.2.1</ip>
             <prefix-length>32</prefix-length>
           </address>
         </ipv4>
         <ipv6 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
           <address>
             <ip>2001:DB8::1</ip>
             <prefix-length>128</prefix-length>
           </address>
         </ipv6>
       </interface>
       <interface>
         <name>Eth1</name>
         <description/>
         <type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type">
         ianaift:ethernetCsmacd
         </type>
         <link-up-down-trap-enable>enabled</link-up-down-trap-enable>
         <ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
           <address>
             <ip>198.51.100.1</ip>
             <prefix-length>30</prefix-length>
           </address>
         </ipv4>
         <ipv6 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
           <address>
             <ip>2001:DB8:0:0:FF::1</ip>
             <prefix-length>64</prefix-length>
           </address>
         </ipv6>
       </interface>
     </interfaces>
   </data>
```


Authors' Addresses

Stephane Litkowski
Cisco Systems

Email: slitkows.ietf@gmail.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

Networking Working Group N. Shen, Ed.
Internet-Draft E. Chen
Intended status: Standards Track A. Lindem
Expires: April 21, 2018 Cisco Systems
October 18, 2017

Carrying Geo Coordinates Information In IS-IS
draft-shen-isis-geo-coordinates-04

Abstract

This document defines a new IS-IS TLV which carries the Geo Coordinates information of the system. The Geo Coordinates information can be used by IS-IS routing or by an application.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The IS-IS routing protocol defined by [ISO10589] has been widely deployed. The Geo Coordinates information can be useful, particularly within the wide area networks for numerous applications. Similar to the Dynamic Hostname defined in [RFC5301], the Geo Coordinates can also be used for network management purposes.

The Geo coordinate information can be retrieve using a variety of means (e.g., SNMP, CLI) without requiring advertising it in an IGP. Nevertheless, announcing the information in IGP allows for new applications and use cases that are elaborated hereafter.

The following provides a non-exhaustive list of sample use cases.

In the case of IGP point-to-multiple operations [I-D.lamparter-isis-p2mp], [RFC6845], the local system configuration can be greatly simplified if the outbound metric to remote neighbors can be generated automatically based on the Geo Location of the IGP neighbors.

In the application where IS-IS neighbors are on the same "sub-net", but over the WAN network, the Geo Location information may be used for equal-cost or unequal-cost load sharing on the local system. This enables location based operation on anycast IP prefixes and DMZ gateways across the WAN environment.

For the traffic matrix using the Geo Coordinates within the routing domain, instead of a collection of IP nexthops which might be translated into locations, this enables automatic region to region traffic pattern aggregation. In particular, introducing new nodes or withdrawing existing ones will be automatically reflected by the application responsible for region to region traffic aggregation. Advanced traffic engineering policies may also be enforced to avoid some nodes located on a specific region under some conditions. Such advanced TE policies are not discussed in this document.

This document describes the IS-IS protocol extension for carrying the Geo Coordinates information. A new TLV is defined for this purpose. This TLV can be distributed within the node’s LSP or inside the IIH PDU. The exact mechanism an application uses the information carried in this TLV is outside the scope of this document.

Further, it is out of scope of this document to specify how a node is provided with the information to be included in the TLV. This document does not assume whether the information included in the TLV is static or not. This is deployment-specific. Typically, this information can be used within a mobile network (trains, for example) that is grafted to a global network.

1.1. 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 RFC 2119 [RFC2119].

2. Packet Encoding

This Geo Coordinates extension introduces one TLV for IS-IS LSP PDU and for Hello (IIH) PDU. The code of the TLV is described in Section 4. The fields specify the location of the system using WGS-84 (World Geodetic System) reference coordinate system [WGS84]. The value of the Geo Coordinates TLV consists of the following fields:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|U|N|E|A|M|R|K|    Reserved     |     Location Uncertainty      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Lat Degrees  |        Latitude Milliseconds                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Long Degrees |        Longitude Milliseconds                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Altitude                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Radius            |          Reserved             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         .. Optional Sub-TLVs
+-+-+-+-+-+-+-+-+-....
```

Type: TBD. 8 bits value, to be assigned by IANA.

Length: Variable. 8 bits value. The mandatory part is 16 octets.

U-bit: If the U-bit is set, it indicates that the "Location Uncertainty" field is specified. If the U-bit is clear, it indicates the "Location Uncertainty" field is unspecified.

N-bit: If the N-bit is set, it indicates the Latitude is north relative to the Equator. If the N-bit is clear, it indicates the Latitude is south of the Equator.

E-bit: If the E-bit is set, it indicates the Longitude is east of the Prime Meridian. If the E-bit is clear, it indicates the Longitude is west of the Prime Meridian.

A-bit: If the A-bit is set, it indicates the "Altitude" field is specified. If the A-bit is clear, it indicates the "Altitude" field is unspecified.

M-bit: If the M-bit is set, it indicates the "Altitude" is specified in meters. If the M-bit is clear, it indicates the "Altitude" is in centimeters.

R-bit: If the R-bit is set, it indicates the "Radius" field is specified and the encoding is for a circular area. If the R-bit is clear, it indicates the "Radius" field is unspecified and the encoding is for a single point.

K-bit: If the K-bit is set, it indicates the "Radius" is specified in kilometers. If the K-bit is clear, it indicates the "Radius" is in meters.

Reserved: These bits are reserved. They SHOULD be set to 0 when sending protocol packets and MUST be ignored when receiving protocol packets.

Location Uncertainty: Unsigned 16-bit integer indicating the number of centimeters of uncertainty for the location.

Latitude Degrees: Unsigned 8-bit integer with a range of 0 - 90 degrees north or south of the Equator (northern or southern hemisphere, respectively).

Latitude Milliseconds: Unsigned 24-bit integer with a range of 0 - 3,599,999 (i.e., less than 60 minutes).

Longitude Degrees: Unsigned 8-bit integer with a range of 0 - 180 degrees east or west of the Prime Meridian.

Longitude Milliseconds: Unsigned 24-bit integer with a range of 0 - 3,599,999 (i.e., less than 60 minutes).

Altitude: Signed 32-bit integer containing the Height relative to sea level in centimeters or meters. A negative height indicates that the location is below sea level.

Radius: Unsigned 16-bit integer containing the radius of a circle centered at the specified coordinates. The radius is specified in meters unless the K-bit is specified indicating specification in kilometers. If the radius is specified, the geo-coordinates specify the entire area of the circle defined by the radius and center point. While the use cases herein do not make use of this field, future use cases may.

Optional Sub-TLV: Not defined in this document, for future extension related to the Geo Coordinates information.

3. Operations

The IS-IS Geo Coordinates TLV may be included in the node’s LSP, and it is recommended to be in the LSP fragment zero. This TLV can also be optionally included in the IIH PDU. This can be useful when the application is setting the outbound p2mp circuit metric based on the neighbor’s location. This can also be used in the Spine-Leaf extension [I-D.shen-isis-spine-leaf-ext] where there is no LSP being flooded into the leaf nodes.

The Geo location information can be provisioned on the system, or it can be dynamically acquired from the GPS capable device on the system.

Further, this specification assumes that the Geo Location coordinates MUST NOT be included by default. An explicit configuration parameter is required to instruct an IS-IS node to include this TLV in its announcement. If a node is instructed to include the TLV, but no value is provided, the TLV MUST NOT be announced.

4. IANA Considerations

A new TLV codepoint is defined in this document and needs to be assigned by IANA from the "IS-IS TLV Codepoints" registry. It is referred to as the Geo Coordinates TLV. This TLV is only to be optionally inserted in the LSP PDU and the IIH PDU. This document does not propose any sub-TLV out of this Geo Coordinates TLV.

Value	Name	IIH	LSP	SNP	Purge
TBD	Geo Coordinates	y	y	n	n

5. Security Considerations

Since the Geo Location coordinates may provide the exact location of the routing devices, disclosure may make the IS-IS devices more susceptible to physical attacks if such IS-IS messages are advertised outside an administrative domain. In situations where this is a concern (e.g., in military applications, or the topology of the network is considered proprietary information), the implementation MUST allow the Geo Location extension to be removed from the IS-IS advertisement. As mentioned in Section 3, the TLV is not included by default. Doing so, allow to avoid misuses of the TLV in the contexts that are not requiring such TLV to be advertised.

Security concerns for the base IS-IS are addressed in [ISO10589], [RFC5304], [RFC5310], and [RFC7602].

6. Privacy Considerations

If the location of an IS-IS router advertising Geo Location coordinates as described herein can be directly correlated to an individual, individuals, or an organization, the location of that router should be considered sensitive and IS-IS LSP containing such geo coordinates should be advertised confidentially as described in Section 5. Additionally, IS-IS network management facilities may require added authorization to view the contents of IS-IS LSPs containing geo-Location TLVs. Refer to [RFC6973] for more information.

The Uncertainty and Confidence metrics for geo-location information as described in [RFC7459] are not included in the Geo Coordinates

TLV. In a future document, these may be considered for inclusion with additional Geo Location Sub-TLVs dependent on both on requirements and adoption of [RFC7459].

7. Acknowledgments

The encoding of the Geo location is adapted from the "Geo Coordinate LISP Canonical Address Format" specified in the "LISP Canonical Address Format (LCAF)". We would like to thank the authors of that Document and particularly Dino Farinacci for subsequent discussions.

Thanks to Mohamed Boucadair, Les Ginsberg, Yi Yang, and Joe Hildebrand for commenting and discussions of Geo Coordinates precision encoding. Thanks to David Ward for commenting on attack vector in relation to this new capability of IS-IS.

8. Document Change Log

8.1. Changes to draft-shen-isis-geo-coordinates-04.txt

- o Clarification and more precise descriptions throughout the document thanks to the detailed comments from Mohamed Boucadair.

8.2. Changes to draft-shen-isis-geo-coordinates-03.txt

- o The 03 version submitted in April 2017 without content change.

8.3. Changes to draft-shen-isis-geo-coordinates-02.txt

- o The 02 version submitted in October 2016.
- o Changed the format of Geo Location encoding to have Radius field and flags to be compatible with LISP [LISP-GEO].
- o Added the privacy section.

8.4. Changes to draft-shen-isis-geo-coordinates-01.txt

- o The 01 version submitted in February 2016.
- o Change Geo Location encoding to have better precision and to include uncertainty information.
- o Added the discussion in security section for the awareness of increased probability in attack vector.

8.5. Changes to draft-shen-isis-geo-coordinates-00.txt

- o Initial version of the draft is published in February 2016.

9. References

9.1. Normative References

[ISO10589]
ISO "International Organization for Standardization", "Intermediate system to Intermediate system intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473), ISO/IEC 10589:2002, Second Edition.", Nov 2002.

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

[RFC5301] McPherson, D. and N. Shen, "Dynamic Hostname Exchange Mechanism for IS-IS", RFC 5301, DOI 10.17487/RFC5301, October 2008, <https://www.rfc-editor.org/info/rfc5301>.

[RFC5304] Li, T. and R. Atkinson, "IS-IS Cryptographic Authentication", RFC 5304, DOI 10.17487/RFC5304, October 2008, <https://www.rfc-editor.org/info/rfc5304>.

[RFC5310] Bhatia, M., Manral, V., Li, T., Atkinson, R., White, R., and M. Fanto, "IS-IS Generic Cryptographic Authentication", RFC 5310, DOI 10.17487/RFC5310, February 2009, <https://www.rfc-editor.org/info/rfc5310>.

[RFC6845] Sheth, N., Wang, L., and J. Zhang, "OSPF Hybrid Broadcast and Point-to-Multipoint Interface Type", RFC 6845, DOI 10.17487/RFC6845, January 2013, <https://www.rfc-editor.org/info/rfc6845>.

[RFC7602] Chunduri, U., Lu, W., Tian, A., and N. Shen, "IS-IS Extended Sequence Number TLV", RFC 7602, DOI 10.17487/RFC7602, July 2015, <https://www.rfc-editor.org/info/rfc7602>.

9.2. Informative References

[I-D.lamparter-isis-p2mp]
Franke, C., Lamparter, D., and C. Hopps, "IS-IS Point-to-Multipoint operation", draft-lamparter-isis-p2mp-01 (work in progress), October 2015.

[I-D.shen-isis-spine-leaf-ext]
Shen, N., Ginsberg, L., and S. Thyamagundalu, "IS-IS Routing for Spine-Leaf Topology", draft-shen-isis-spine-leaf-ext-03 (work in progress), March 2017.

[LISP-GEO]
Farinacci, D., "LISP Geo-Coordinate Use-Cases", draft-farinacci-lisp-geo-02 (work in progress), 2016.

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <https://www.rfc-editor.org/info/rfc6973>.

[RFC7459] Thomson, M. and J. Winterbottom, "Representation of Uncertainty and Confidence in the Presence Information Data Format Location Object (PIDF-LO)", RFC 7459, DOI 10.17487/RFC7459, February 2015, <https://www.rfc-editor.org/info/rfc7459>.

[WGS84] National Imagery and Mapping Agency, "Department of Defense World Geodetic System 1984, Third Edition", NIMA TR8350.2, January 2000.

Authors' Addresses

Naiming Shen (editor)
Cisco Systems
560 McCarthy Blvd.
Milpitas, CA 95035
US

Email: naiming@cisco.com

Enke Chen
Cisco Systems
560 McCarthy Blvd.
Milpitas, CA 95035
US

Email: enkechen@cisco.com

Acee Linden
Cisco Systems
301 Midenhall Way
Cary, NC 27513
US

Email: acee@cisco.com

Networking Working Group N. Shen
Internet-Draft L. Ginsberg
Intended status: Standards Track Cisco Systems
Expires: April 19, 2019 S. Thyamagundalu
October 16, 2018

IS-IS Routing for Spine-Leaf Topology
draft-shen-isis-spine-leaf-ext-07

Abstract

This document describes a mechanism for routers and switches in a Spine-Leaf type topology to have non-reciprocal Intermediate System to Intermediate System (IS-IS) routing relationships between the leafs and spines. The leaf nodes do not need to have the topology information of other nodes and exact prefixes in the network. This extension also has application in the Internet of Things (IoT).

Table of Contents

1. Introduction

The IS-IS routing protocol defined by [ISO10589] has been widely deployed in provider networks, data centers and enterprise campus environments. In the data center and enterprise switching networks, a Spine-Leaf topology is commonly used. This document describes a mechanism where IS-IS routing can be optimized for a Spine-Leaf topology.

In a Spine-Leaf topology, normally a leaf node connects to a number of spine nodes. Data traffic going from one leaf node to another leaf node needs to pass through one of the spine nodes. Also, the decision to choose one of the spine nodes is usually part of equal cost multi-path (ECMP) load sharing. The spine nodes can be considered as gateway devices to reach destinations on other leaf nodes. In this type of topology, the spine nodes have to know the topology and routing information of the entire network, but the leaf nodes only need to know how to reach the gateway devices to which are the spine nodes they are uplinked.

This document describes the IS-IS Spine-Leaf extension that allows the spine nodes to have all the topology and routing information, while keeping the leaf nodes free of topology information other than the default gateway routing information. The leaf nodes do not even need to run a Shortest Path First (SPF) calculation since they have no topology information.

1.1. 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 RFC 2119 [RFC2119].

2. Motivations

- o The leaf nodes in a Spine-Leaf topology do not require complete topology and routing information of the entire domain since their forwarding decision is to use ECMP with spine nodes as default gateways

- o The spine nodes in a Spine-Leaf topology are richly connected to leaf nodes, which introduces significant flooding duplication if they flood all Link State PDUs (LSPs) to all the leaf nodes. It saves both spine and leaf nodes’ CPU and link bandwidth resources if flooding is blocked to leaf nodes. For small Top of the Rack (ToR) leaf switches in data centers, it is meaningful to prevent full topology routing information and massive database flooding through those devices.

- o When a spine node advertises a topology change, every leaf node connected to it will flood the update to all the other spine nodes, and those spine nodes will further flood them to all the leaf nodes, causing a O(n^2) flooding storm which is largely redundant.

- o Similar to some of the overlay technologies which are popular in data centers, the edge devices (leaf nodes) may not need to contain all the routing and forwarding information on the device’s control and forwarding planes. "Conversational Learning" can be utilized to get the specific routing and forwarding information in the case of pure CLOS topology and in the events of link and node down.

- o Small devices and appliances of Internet of Things (IoT) can be considered as leafs in the routing topology sense. They have CPU and memory constrains in design, and those IoT devices do not have to know the exact network topology and prefixes as long as there are ways to reach the cloud servers or other devices.

3. Spine-Leaf (SL) Extension

3.1. Topology Examples

```
             +--------+    +--------+             +--------+
             |        |    |        |             |        |
             | Spine1 +----+ Spine2 +- ......... -+ SpineN |
             |        |    |        |             |        |
             +-+-+-+-++    ++-+-+-+-+             +-+-+-+-++
        +------+ | | |      | | | |                 | | | |
        |  +-----|-|-|------+ | | |                 | | | |
        |  |  +--|-|-|--------+-|-|-----------------+ | | |
        |  |  |  | | |    +---+ | |                   | | |
        |  |  |  | | |    |  +--|-|-------------------+ | |
        |  |  |  | | |    |  |  | |              +------+ +----+
        |  |  |  | | |    |  |  | +--------------|----------+  |
        |  |  |  | | |    |  |  +-------------+  |          |  |
        |  |  |  | | +----|--|----------------|--|--------+ |  |
        |  |  |  | +------|--|--------------+ |  |        | |  |
        |  |  |  +------+ |  |              | |  |        | |  |
       ++--+--++      +-+-+--++            ++-+--+-+     ++-+--+-+
       | Leaf1 |      | Leaf2 |  ........  | LeafX |     | LeafY |
       +-------+      +-------+            +-------+     +-------+
```

Figure 1: A Spine-Leaf Topology

```
                 +---------+            +--------+
                 | Spine1  |            | Spine2 |
                 +-+-+-+-+-+            +-+-+-+-++
                   | | | |                | | | |
                   | | | +----------------|-|-|-|-+
                   | | +-----------+      | | | | |
          +--------+ +-+           |      | | | | |
          |   +---------------------------+ | | | |
          |   |        |  +-----------------+ | +----+
          |   |        |  |        |  +-------+   |  |
          |   |        |  |        |  |           |  |
        +-+---+-+   +--+--+-+    +-+--+--+     +--+--+-+
        | Leaf1 |   | Leaf2 |    | Leaf3 |     | Leaf4 |
        +-------+   +-------+    +-------+     +-------+
```

Figure 2: A CLOS Topology

3.2. Applicability Statement

This extension assumes the network is a Spine-Leaf topology, and it should not be applied in an arbitrary network setup. The spine nodes can be viewed as the aggregation layer of the network, and the leaf nodes as the access layer of the network. The leaf nodes use a load sharing algorithm with spine nodes as nexthops in routing and forwarding.

This extension works when the spine nodes are inter-connected, and it works with a pure CLOS or Fat Tree topology based network where the spines are NOT horizontally interconnected.

Although the example diagram in Figure 1 shows a fully meshed Spine-Leaf topology, this extension also works in the case where they are partially meshed. For instance, leaf1 through leaf10 may be fully meshed with spine1 through spine5 while leaf11 through leaf20 is fully meshed with spine4 through spine8, and all the spines are inter-connected in a redundant fashion.

This extension can also work in multi-level spine-leaf topology. The lower level spine node can be a 'leaf' node to the upper level spine node. A spine-leaf 'Tier' can be exchanged with IS-IS hello packets to allow tier X to be connected with tier X+1 using this extension. Normally tier-0 will be the TOR routers and switches if provisioned.

This extension also works with normal IS-IS routing in a topology with more than two layers of spine and leaf. For instance, in example diagrams Figure 1 and Figure 2, there can be another Core layer of routers/switches on top of the aggregation layer. From an IS-IS routing point of view, the Core nodes are not affected by this extension and will have the complete topology and routing information just like the spine nodes. To make the network even more scalable, the Core layer can operate as a level-2 IS-IS sub-domain while the Spine and Leaf layers operate as stays at the level-1 IS-IS domain.

This extension assumes the link between the spine and leaf nodes are point-to-point, or point-to-point over LAN [RFC5309]. The links connecting among the spine nodes or the links between the leaf nodes can be any type.

3.3. Spine-Leaf TLV

This extension introduces a new TLV, the Spine-Leaf TLV, which may be advertised in IS-IS Hello (IIH) PDUs, LSPs, or in Circuit Scoped Link State PDUs (CS-LSP) [RFC7356]. It is used by both spine and leaf nodes in this Spine-Leaf mechanism.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Type     |     Length    |            SL Flag            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         .. Optional Sub-TLVs
+-+-+-+-+-+-+-+-+-....
```

The fields of this TLV are defined as follows:

Type: 1 octet Suggested value 150 (to be assigned by IANA)

Length: 1 octet (2 + length of sub-TLVs).

SL Flags: 16 bits

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Tier |     Reserved    |T|R|L|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Tier: A value from 0 to 15. It represents the spine-leaf tier level. The value 15 is reserved to indicate the tier level is unknown. This value is only valid when the 'T' bit (see below) is set. If the 'T' bit is clear, this value MUST be set to zero on transmission, and it MUST be ignored on receipt.

L bit (0x01): Only leaf node sets this bit. If the L bit is set in the SL flag, the node indicates it is in 'Leaf-Mode'.

R bit (0x02): Only Spine node sets this bit. If the R bit is set, the node indicates to the leaf neighbor that it can be used as the default route gateway.

T bit (0x04): If set, the value in the "Tier" field (see above) is valid.

Optional Sub-TLV: Not defined in this document, for future extension

sub-TLVs MAY be included when the TLV is in a CS-LSP. sub-TLVs MUST NOT be included when the TLV is in an IIH

3.3.1. Spine-Leaf Sub-TLVs

If the data center topology is a pure CLOS or Fat Tree, there are no link connections among the spine nodes. If we also assume there is not another Core layer on top of the aggregation layer, then the traffic from one leaf node to another may have a problem if there is a link outage between a spine node and a leaf node. For instance, in the diagram of Figure 2, if Leaf1 sends data traffic to Leaf3 through Spine1 node, and the Spine1-Leaf3 link is down, the data traffic will be dropped on the Spine1 node.

To address this issue spine and leaf nodes may send/request specific reachability information via the sub-TLVs defined below.

Two Spine-Leaf sub-TLVs are defined. The Leaf-Set sub-TLV and the Info-Req sub-TLV.

3.3.1.1. Leaf-Set Sub-TLV

This sub-TLV is used by spine nodes to optionally advertise Leaf neighbors to other Leaf nodes. The fields of this sub-TLV are defined as follows:

Type: 1 octet Suggested value 1 (to be assigned by IANA)

Length: 1 octet MUST be a multiple of 6 octets.

Leaf-Set: A list of IS-IS System-ID of the leaf node neighbors of this spine node.

3.3.1.2. Info-Req Sub-TLV

This sub-TLV is used by leaf nodes to request the advertisement of more specific prefix information from a selected spine node. The list of leaf nodes in this sub-TLV reflects the current set of leaf-nodes for which not all spine node neighbors have indicated the presence of connectivity in the Leaf-Set sub-TLV (See Section 3.3.1.1). The fields of this sub-TLV are defined as follows:

Type: 1 octet Suggested value 2 (to be assigned by IANA)

Length: 1 octet. It MUST be a multiple of 6 octets.

Info-Req: List of IS-IS System-IDs of leaf nodes for which connectivity information is being requested.

3.3.2. Advertising IPv4/IPv6 Reachability

In cases where connectivity between a leaf node and a spine node is down, the leaf node MAY request reachability information from a spine node as described in Section 3.3.1.2. The spine node utilizes TLVs 135 [RFC5305] and TLVs 236 [RFC5308] to advertise this information. These TLVs MAY be included either in IIHs or CS-LSPs [RFC7356] sent from the spine to the requesting leaf node. Sending such information in IIHs has limited scale - all reachability information MUST fit within a single IIH. It is therefore recommended that CS-LSPs be used.

3.3.3. Advertising Connection to RF-Leaf Node

For links between Spine and Leaf Nodes on which the Spine Node has set the R-bit and the Leaf node has set the L-bit in their respective Spine-Leaf TLVs, spine nodes may advertise the link with a bit in the "link-attribute" sub-TLV [RFC5029] to express this link is not used for LSP flooding. This information can be used by nodes computing a flooding topology e.g., [DYNAMIC-FLOODING], to exclude the RF-Leaf nodes from the computed flooding topology.

3.4. Mechanism

Leaf nodes in a spine-leaf application using this extension are provisioned with two attributes:

1)Tier level of 0. This indicates the node is a Leaf Node. The value 0 is advertised in the Tier field of Spine-Leaf TLV defined above.

2)Flooding reduction enabled/disabled. If flooding reduction is enabled the L-bit is set to one in the Spine-Leaf TLV defined above

A spine node does not need explicit configuration. Spine nodes can dynamically discover their tier level by computing the number of hops to a leaf node. Until a spine node determines its tier level it MUST advertise level 15 (unknown tier level) in the Spine-Leaf TLV defined above. Each tier level can also be statically provisioned on the node.

When a spine node receives an IIH which includes the Spine-Leaf TLV with Tier level 0 and 'L' bit set, it labels the point-to-point interface and adjacency to be a 'Reduced Flooding Leaf-Peer (RF-Leaf)'. IIHs sent by a spine node on a link to an RF-Leaf include the Spine-Leaf TLV with the 'R' bit set in the flags field. The 'R' bit indicates to the RF-Leaf neighbor that the spine node can be used as a default routing nexthop.

There is no change to the IS-IS adjacency bring-up mechanism for Spine-Leaf peers.

A spine node blocks LSP flooding to RF-Leaf adjacencies, except for the LSP PDUs in which the IS-IS System-ID matches the System-ID of the RF-Leaf neighbor. This exception is needed since when the leaf node reboots, the spine node needs to forward to the leaf node non-purged LSPs from the RF-Leaf's previous incarnation.

Leaf nodes will perform IS-IS LSP flooding as normal over all of its IS-IS adjacencies, but in the case of RF-Leafs only self-originated LSPs will exist in its LSP database.

Spine nodes will receive all the LSP PDUs in the network, including all the spine nodes and leaf nodes. It will perform Shortest Path First (SPF) as a normal IS-IS node does. There is no change to the route calculation and forwarding on the spine nodes.

The LSPs of a node only floods north bound towards the upper layer spine nodes. The default route is generated with loadsharing also towards the upper layer spine nodes.

RF-Leaf nodes do not have any LSP in the network except for its own. Therefore there is no need to perform SPF calculation on the RF-Leaf node. It only needs to download the default route with the nexthops of those Spine Neighbors which have the 'R' bit set in the Spine-Leaf TLV in IIH PDUs. IS-IS can perform equal cost or unequal cost load sharing while using the spine nodes as nexthops. The aggregated metric of the outbound interface and the 'Reverse Metric' [REVERSE-METRIC] can be used for this purpose.

3.4.1. Pure CLOS Topology

In a data center where the topology is pure CLOS or Fat Tree, there is no interconnection among the spine nodes, and there is not another Core layer above the aggregation layer with reachability to the leaf nodes. When flooding reduction to RF-Leafs is in use, if the link between a spine and a leaf goes down, there is then a possibility of black holing the data traffic in the network.

As in the diagram Figure 2, if the link Spine1-Leaf3 goes down, there needs to be a way for Leaf1, Leaf2 and Leaf4 to avoid the Spine1 if the destination of data traffic is to Leaf3 node.

In the above example, the Spine1 and Spine2 are provisioned to advertise the Leaf-Set sub-TLV of the Spine-Leaf TLV. Originally both Spines will advertise Leaf1 through Leaf4 as their Leaf-Set. When the Spine1-Leaf3 link is down, Spine1 will only have Leaf1, Leaf2 and Leaf4 in its Leaf-Set. This allows the other leaf nodes to know that Spine1 has lost connectivity to the leaf node of Leaf3.

Each RF-Leaf node can select another spine node to request for some prefix information associated with the lost leaf node. In this diagram of Figure 2, there are only two spine nodes (Spine-Leaf topology can have more than two spine nodes in general). Each RF-Leaf node can independently select a spine node for the leaf information. The RF-Leaf nodes will include the Info-Req sub-TLV in the Spine-Leaf TLV in hellos sent to the selected spine node, Spine2 in this case.

The spine node, upon receiving the request from one or more leaf nodes, will find the IPv6/IPv4 prefixes advertised by the leaf nodes listed in the Info-Req sub-TLV. The spine node will use the mechanism defined in Section 3.3.2 to advertise these prefixes to the RF-Leaf node. For instance, it will include the IPv4 loopback prefix of leaf3 based on the policy configured or administrative tag attached to the prefixes. When the leaf nodes receive the more specific prefixes, they will install the advertised prefixes towards the other spine nodes (Spine2 in this example).

For instance in the data center overlay scenario, when any IP destination or MAC destination uses the leaf3’s loopback as the tunnel nexthop, the overlay tunnel from leaf nodes will only select Spine2 as the gateway to reach leaf3 as long as the Spine1-Leaf3 link is still down.

In cases where multiple links or nodes fail at the same time, the RF-leaf node may need to send the Info-Req to multiple upper layer spine nodes in order to obtain reachability information for all the partially connected nodes.

This negative routing is more useful between tier 0 and tier 1 spine-leaf levels in a multi-level spine-leaf topology when the reduced flooding extension is in use. Nodes in tiers 1 or greater may have much richer topology information and alternative paths.

3.5. Implementation and Operation

3.5.1. CSNP PDU

In Spine-Leaf extension, Complete Sequence Number PDU (CSNP) does not need to be transmitted over the Spine-Leaf link to an RF-Leaf. Some IS-IS implementations send periodic CSNPs after the initial adjacency bring-up over a point-to-point interface. There is no need for this optimization here since the RF-Leaf does not need to receive any other LSPs from the network, and the only LSPs transmitted across the Spine-Leaf link is the leaf node LSP.

Also in the graceful restart case[RFC5306], for the same reason, there is no need to send the CSNPs over the Spine-Leaf interface to an RF-Leaf. Spine nodes only need to set the SRMflag on the LSPs belonging to the RF-Leaf.

3.5.2. Overload Bit

The leaf node SHOULD set the 'overload' bit on its LSP PDU, since if the spine nodes were to forward traffic not meant for the local node, the leaf node does not have the topology information to prevent a routing/forwarding loop.

3.5.3. Spine Node Hostname

This extension creates a non-reciprocal relationship between the spine node and leaf node. The spine node will receive leaf's LSP and will know the leaf's hostname, but the leaf does not have spine's LSP. This extension allows the Dynamic Hostname TLV [RFC5301] to be optionally included in spine's IIH PDU when sending to a 'Leaf-Peer'. This is useful in troubleshooting cases.

3.5.4. IS-IS Reverse Metric

This metric is part of the aggregated metric for leaf's default route installation with load sharing among the spine nodes. When a spine node is in 'overload' condition, it should use the IS-IS Reverse Metric TLV in IIH [REVERSE-METRIC] to set this metric to maximum to discourage the leaf using it as part of the loadsharing.

In some cases, certain spine nodes may have less bandwidth in link provisioning or in real-time condition, and it can use this metric to signal to the leaf nodes dynamically.

In other cases, such as when the spine node loses a link to a particular leaf node, although it can redirect the traffic to other spine nodes to reach that destination leaf node, but it MAY want to increase this metric value if the inter-spine connection becomes over utilized, or the latency becomes an issue.

In the leaf-leaf link as a backup gateway use case, the 'Reverse Metric' SHOULD always be set to very high value.

3.5.5. Spine-Leaf Traffic Engineering

Besides using the IS-IS Reverse Metric by the spine nodes to affect the traffic pattern for leaf default gateway towards multiple spine nodes, the IPv6/IPv4 Info-Advertise sub-TLVs can be selectively used by traffic engineering controllers to move data traffic around the data center fabric to alleviate congestion and to reduce the latency of a certain class of traffic pairs. By injecting more specific leaf node prefixes, it will allow the spine nodes to attract more traffic on some underutilized links.

3.5.6. Other End-to-End Services

Losing the topology information will have an impact on some of the end-to-end network services, for instance, MPLS TE or end-to-end segment routing. Some other mechanisms such as those described in PCE [RFC4655] based solution may be used. In this Spine-Leaf extension, the role of the leaf node is not too much different from the multi-level IS-IS routing while the level-1 IS-IS nodes only have the default route information towards the node which has the Attach Bit (ATT) set, and the level-2 backbone does not have any topology information of the level-1 areas. The exact mechanism to enable certain end-to-end network services in Spine-Leaf network is outside the scope of this document.

3.5.7. Address Family and Topology

IPv6 Address families[RFC5308], Multi-Topology (MT)[RFC5120] and Multi-Instance (MI)[RFC8202] information is carried over the IIH PDU. Since the goal is to simplify the operation of IS-IS network, for the simplicity of this extension, the Spine-Leaf mechanism is applied the same way to all the address families, MTs and MIs.

3.5.8. Migration

For this extension to be deployed in existing networks, a simple migration scheme is needed. To support any leaf node in the network, all the involved spine nodes have to be upgraded first. So the first step is to migrate all the involved spine nodes to support this extension, then the leaf nodes can be enabled with 'Leaf-Mode' one by one. No flag day is needed for the extension migration.

4. IANA Considerations

A new TLV codepoint is defined in this document and needs to be assigned by IANA from the "IS-IS TLV Codepoints" registry. It is referred to as the Spine-Leaf TLV and the suggested value is 150. This TLV is only to be optionally inserted either in the IIH PDU or in the Circuit Flooding Scoped LSP PDU. IANA is also requested to maintain the SL-flag bit values in this TLV, and 0x01, 0x02 and 0x04 bits are defined in this document.

Value	Name	IIH	LSP	SNP	Purge	CS-LSP
150	Spine-Leaf	y	y	n	n	y

This extension also proposes to have the Dynamic Hostname TLV, already assigned as code 137, to be allowed in IIH PDU.

Value	Name	IIH	LSP	SNP	Purge
137	Dynamic Name	y	y	n	y

Two new sub-TLVs are defined in this document and needs to be added assigned by IANA from the "IS-IS TLV Codepoints". They are referred to in this document as the Leaf-Set sub-TLV and the Info-Req sub-TLV. It is suggested to have the values 1 and 2 respectively.

This document also requests that IANA allocate from the registry of link-attribute bit values for sub-TLV 19 of TLV 22 (Extended IS reachability TLV). This new bit is referred to as the "Connect to RF-Leaf Node" bit.

Value	Name	Reference
0x3	Connect to RF-Leaf Node	This document

5. Security Considerations

Security concerns for IS-IS are addressed in [ISO10589], [RFC5304], [RFC5310], and [RFC7602]. This extension does not raise additional security issues.

6. Acknowledgments

The authors would like to thank Tony Przygienda for his discussion and contributions. The authors also would like to thank Acee Lindem, Russ White and Christian Hopps for their review and comments of this document.

7. Document Change Log

7.1. Changes to draft-shen-isis-spine-leaf-ext-05.txt

- Submitted January 2018.
- Just a refresh.

7.2. Changes to draft-shen-isis-spine-leaf-ext-04.txt

- Submitted June 2017.
- Added the Tier level information to handle the multi-level spine-leaf topology using this extension.

7.3. Changes to draft-shen-isis-spine-leaf-ext-03.txt

- Submitted March 2017.
- Added the Spine-Leaf sub-TLVs to handle the case of data center pure CLOS topology and mechanism.
- Added the Spine-Leaf TLV and sub-TLVs can be optionally inserted in either IIH PDU or CS-LSP PDU.
- Allow use of prefix Reachability TLVs 135 and 236 in IIHs/CS-LSPs sent from spine to leaf.

7.4. Changes to draft-shen-isis-spine-leaf-ext-02.txt

- Submitted October 2016.
- Removed the ’Default Route Metric’ field in the Spine-Leaf TLV and changed to using the IS-IS Reverse Metric in IIH.

7.5. Changes to draft-shen-isis-spine-leaf-ext-01.txt

- Submitted April 2016.
- No change. Refresh the draft version.

7.6. Changes to draft-shen-isis-spine-leaf-ext-00.txt

- Initial version of the draft is published in November 2015.

8. References

8.1. Normative References

[ISO10589]
ISO "International Organization for Standardization", "Intermediate system to Intermediate system intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473), ISO/IEC 10589:2002, Second Edition.", Nov 2002.

[REVERSE-METRIC]
Shen, N., Amante, S., and M. Abrahamsson, "IS-IS Routing with Reverse Metric", draft-ietf-isis-reverse-metric-07 (work in progress), 2017.

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

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

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

[RFC5301] McPherson, D. and N. Shen, "Dynamic Hostname Exchange Mechanism for IS-IS", RFC 5301, DOI 10.17487/RFC5301, October 2008, <https://www.rfc-editor.org/info/rfc5301>.

[RFC5304] Li, T. and R. Atkinson, "IS-IS Cryptographic Authentication", RFC 5304, DOI 10.17487/RFC5304, October 2008, <https://www.rfc-editor.org/info/rfc5304>.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <https://www.rfc-editor.org/info/rfc5305>.

[RFC5306] Shand, M. and L. Ginsberg, "Restart Signaling for IS-IS", RFC 5306, DOI 10.17487/RFC5306, October 2008, <https://www.rfc-editor.org/info/rfc5306>.

[RFC5308] Hopps, C., "Routing IPv6 with IS-IS", RFC 5308, DOI 10.17487/RFC5308, October 2008, <https://www.rfc-editor.org/info/rfc5308>.

[RFC5310] Bhatia, M., Manral, V., Li, T., Atkinson, R., White, R., and M. Fanto, "IS-IS Generic Cryptographic Authentication", RFC 5310, DOI 10.17487/RFC5310, February 2009, <https://www.rfc-editor.org/info/rfc5310>.

[RFC7356] Ginsberg, L., Previdi, S., and Y. Yang, "IS-IS Flooding Scope Link State PDUs (LSPs)", RFC 7356, DOI 10.17487/RFC7356, September 2014, <https://www.rfc-editor.org/info/rfc7356>.

[RFC7602] Chunduri, U., Lu, W., Tian, A., and N. Shen, "IS-IS Extended Sequence Number TLV", RFC 7602, DOI 10.17487/RFC7602, July 2015, <https://www.rfc-editor.org/info/rfc7602>.

[RFC8202] Ginsberg, L., Previdi, S., and W. Henderickx, "IS-IS Multi-Instance", RFC 8202, DOI 10.17487/RFC8202, June 2017, <https://www.rfc-editor.org/info/rfc8202>.

8.2. Informative References

[DYNAMIC-FLOODING] Li, T., "Dynamic Flooding on Dense Graphs", draft-li-dynamic-flooding (work in progress), 2018.

[RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <https://www.rfc-editor.org/info/rfc4655>.

[RFC5309] Shen, N., Ed. and A. Zinin, Ed., "Point-to-Point Operation over LAN in Link State Routing Protocols", RFC 5309, DOI 10.17487/RFC5309, October 2008, <https://www.rfc-editor.org/info/rfc5309>.

Authors' Addresses

Naiming Shen
Cisco Systems
560 McCarthy Blvd.
Milpitas, CA 95035
US

Email: naiming@cisco.com

Les Ginsberg
Cisco Systems
821 Alder Drive
Milpitas, CA 95035
US

Email: ginsberg@cisco.com

Sanjay Thyamagundalu

Email: tsanjay@gmail.com

Network Working Group X. Xu
Internet-Draft Huawei
Intended status: Standards Track S. Dikshit
Expires: February 9, 2017 Cisco
H. Shah
Ciena Corp
Y. Fan
China Telecom
August 8, 2016

NVO Control Plane Protocol Using IS-IS
draft-xu-isis-nvo-cp-00

Abstract

This document describes the use of IS-IS as a light-weight control plane protocol for Network Virtualization Overlays. This light-weight control plane protocol is intended for small and even medium sized enterprise campus networks where the NVO date encapsulation technology is to be used.

Table of Contents

1. Introduction

[RFC7364] discusses the need of an overlay-based network virtualization approach, referred to as Network Virtualization Overlays (NVO), for providing multi-tenancy capabilities in large data centers networks and outlines the needs for a control plane protocol to facilitate running NVO. [RFC7365] provides a framework for NVO and meanwhile describes the needs for a control plane protocol to provide the following capabilities such as auto-provisioning/service discovery, address mapping advertisement and tunnel management.

Due to the success of the NVO technology in data center networks, more and more enterprises are considering the deployment of this technology in their campus networks so as to replace the old spanning tree protocols. Although BGP or Software Defined Network (SDN) controller could still be used as the control plane protocol in campus networks, both of them seem a bit heavyweight, especially for small and even medium sized campus networks.

IS-IS protocol [IS-IS] is a much proven and well-known routing protocol which has been widely deployed in campus networks for many years. Due to its extendibility, IS-IS protocol now is not only used for propagating IP reachability information in Layer3 networks (see [RFC1195]), but also used for propagating MAC reachability information in Layer2 networks or Layer2 overlay networks [RFC6165].

By using IS-IS as a lightweight control plane protocol for NVO, the network provisioning is greatly simplified ((e.g., only a single protocol to be deployed)), which is much significant to campus networks.

This IS-IS based NVO control plane protocol could support any specific NVO data encapsulation formats such as VXLAN [RFC7348], VXLAN-GPE [I-D.ietf-nvo3-vxlan-gpe] , and NVGRE [RFC7637].

1.1. 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 RFC 2119 [RFC2119].

2. Terminology

This memo makes use of the terms defined in [RFC7365] and [I-D.ietf-bier-architecture].

3. VN Membership Auto-discovery

By propagating the VN membership info among Network Virtualization Edges (NVEs), NVEs belonging to the same VN instance could discover one another automatically. The VN membership info is carried in a VN Membership Info sub-TLV (as shown in Section 3.1) of the following TLVs originated by that NVE:

1. TLV-135 (IPv4) defined in [RFC5305].

2. TLV-236 (IPv6) defined in [RFC5308]

When the above TLV is propagated across level boundaries, the VN Membership Info sub-TLV contained in that TLV SHOULD be kept.

3.1. VN Membership Info Sub-TLV

The VN Membership Info sub-TLV has the following format:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Type=TBD   |     Length    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  VN ID                        |S|   Reserved  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Sub-domain ID |                     Reserved                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  VN ID                        |S|   Reserved  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Sub-domain ID |                     Reserved                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type: TBD;

Length: Variable;

VN ID: This field is filled with a 24-bit globally significant VN ID for a particular attached VN instance.

S-Flag: This field indicates the existence of the Sub-domain ID field. When the S-Flag is set, the Sub-domain ID field MUST be filled with a valid sub-domain ID. Otherwise, it SHOULD be set to zero.

Sub-domain ID: This field is filled with a 8-bit BIER sub-domain ID to which the VN has been associated [I-D.ietf-bier-architecture]. The field is only useful in the case where the Broadcast, Unknown-unicast and Multicast (BUM) packets within a VN are transported across the underlay by using the BIER forwarding mode.

4. Tunnel Encapsulation Capability Advertisement

To reach a consensus on what specific tunnel encapsulation format to be used between ingress and egress NVE pairs automatically, egress NVEs SHOULD advertise their own tunnel encapsulation capabilities by using the Encapsulation Capability sub-TLV as defined in [I-D.xu-isis-encapsulation-cap]

5. MAC Address Learning

For Layer2 overlays, MAC addresses of local CE hosts would still be learnt by NVEs as normal bridges. As for learning MAC addresses of remote CE hosts, there are two options: 1) data-plane based MAC learning and 2) control- plane based MAC learning. If unknown unicast flood suppression is strongly required even at the cost of consuming more forwarding table resources, the control-plane based MAC learning option could be considered. Otherwise, the data-plane based MAC learning option is RECOMMENDED.

5.1. Control-plane based MAC Learning for Remote CE Hosts

In the control-plane based MAC address learning mechanism, MAC reachability information of a given VN instance would be exchanged across NVEs of that VN instance via IS-IS as well. Upon learning MAC addresses of their local TES’s somehow, NVEs SHOULD immediately advertise these MAC addresses to remote NVEs of the same VN instance by using the MAC-Reachability TLV as defined in [RFC6165]. One or more MAC-Reachability TLVs are carried in an LSP which in turn is encapsulated with an Ethernet header. The source MAC address is the originating NVE’s MAC address whereas the destination MAC address is a to-be-defined multicast MAC address specifically identifying all NVEs. Although in Ingress Replication case for networks not supporting multicast, the remote NVE unicast addresses can be pre-learned via configuration, and used as destination MAC address instead of multicast MAC address. Such Ethernet frames containing IS-IS LSPs are forwarded towards remote NVEs as if they were customer multicast Ethernet frames. Egress NVEs receiving the above frames SHOULD intercept them and accordingly process them. The routable IP address of the NVE originating these MAC routes could be derived either from the "IP Interface Address" field contained in the corresponding LSPs (Note that the IP address here SHOULD be identical to the routable IP address associated with the VN membership Info) or from the tunnel source IP address of the NVO encapsulated packet containing such MAC routes. Since these LSPs are fully transparent to core routers of the underlying networks (i.e., non-NVE routers), there is no impact on the control plane of core routers at all.

6. MAC/IP Binding Info Advertisement

To refrain from flooding ARP/ND messages generated by end-hosts, across all NVEs for a given VN, IP/MAC bindings for these end-hosts can be potentially exchanged between NVEs through IS-IS. ARP/ND caching can be enabled on NVEs to allow local NVE to respond for an ARP/ND requests on behalf of remote hosts. Thus there is no need to flood ARP/ND messages to all other NVEs of a given VN. This potential extension is for further study

7. IP Reachability Info Advertisement

For Layer3 overlays, IP reachability information of a given VN instance, including both host routes and/or subnet routes, SHOULD be exchanged across NVEs of that VN instance. The IP-Reachability TLV defined in [RFC1195] could be used directly here. One or more IP-Reachability TLVs are carried in a LSP which in turn is encapsulated with an Ethernet header. The source MAC address is the originating NVE's MAC address whereas the destination MAC address is a to-be-defined multicast MAC address specifically identifying all NVEs. Such Ethernet frames containing IS-IS LSPs are forwarded towards remote NVEs as if they were customer multicast Ethernet frames. Egress NVEs receiving the above frames SHOULD intercept them and accordingly process them. Similarly, since these LSPs are fully transparent to core routers of the underlying networks (i.e., non-NVE routers), there is no impact on the control plane of core routers at all.

8. IANA Considerations

The type code for VN Membership Info sub-TLV is required to be allocated by IANA.

9. Security Considerations

This document doesn't introduce additional security risk to IS-IS, nor does it provide any additional security feature for IS-IS.

10. Acknowledgements

TBD

11. References

11.1. Normative References

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

[RFC4971] Vasseur, JP., Ed., Shen, N., Ed., and R. Aggarwal, Ed., "Intermediate System to Intermediate System (IS-IS) Extensions for Advertising Router Information", RFC 4971, DOI 10.17487/RFC4971, July 2007, <http://www.rfc-editor.org/info/rfc4971>.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <http://www.rfc-editor.org/info/rfc5305>.

[RFC5308] Hopps, C., "Routing IPv6 with IS-IS", RFC 5308, DOI 10.17487/RFC5308, October 2008, <http://www.rfc-editor.org/info/rfc5308>.

11.2. Informative References

[I-D.ietf-bier-architecture]
Wijnands, I., Rosen, E., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast using Bit Index Explicit Replication", draft-ietf-bier-architecture-04 (work in progress), July 2016.

[I-D.ietf-nvo3-vxlan-gpe]
Kreeger, L. and U. Elzur, "Generic Protocol Extension for VXLAN", draft-ietf-nvo3-vxlan-gpe-02 (work in progress), April 2016.

[I-D.xu-isis-encapsulation-cap]
Xu, X., Decraene, B., Raszuk, R., Chunduri, U., Contreras, L., and L. Jalil, "Advertising Tunnelling Capability in IS-IS", draft-xu-isis-encapsulation-cap-06 (work in progress), November 2015.

[IS-IS] "ISO/IEC 10589, "Intermediate System to Intermediate System Intra-Domain Routing Exchange Protocol for use in Conjunction with the Protocol for Providing the Connectionless-mode Network Service (ISO 8473)", 2005.".

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

[RFC6165] Banerjee, A. and D. Ward, "Extensions to IS-IS for Layer-2 Systems", RFC 6165, DOI 10.17487/RFC6165, April 2011, <http://www.rfc-editor.org/info/rfc6165>.

[RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <http://www.rfc-editor.org/info/rfc7348>.

[RFC7364] Narten, T., Ed., Gray, E., Ed., Black, D., Fang, L., Kreeger, L., and M. Napierala, "Problem Statement: Overlays for Network Virtualization", RFC 7364, DOI 10.17487/RFC7364, October 2014, <http://www.rfc-editor.org/info/rfc7364>.

[RFC7365] Lasserre, M., Balus, F., Morin, T., Bitar, N., and Y. Rekhter, "Framework for Data Center (DC) Network Virtualization", RFC 7365, DOI 10.17487/RFC7365, October 2014, <http://www.rfc-editor.org/info/rfc7365>.

[RFC7637] Garg, P., Ed. and Y. Wang, Ed., "NVGRE: Network Virtualization Using Generic Routing Encapsulation", RFC 7637, DOI 10.17487/RFC7637, September 2015, <http://www.rfc-editor.org/info/rfc7637>.

Authors' Addresses

Xiaohu Xu
Huawei

Email: xuxiaohu@huawei.com

Saumya Dikshit
Cisco

Email: sadikshi@cisco.com

Himanshu Shah
Ciena Corp

Email: hshah@ciena.com

Yongbing Fan
China Telecom

Email: fanyb@gsta.com