

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: February 19, 2021

H. Chen
R. Li
Futurewei
Y. Yang
IBM
A. Kumar S N
RtBrick
Y. Fan
Casa Systems
N. So

V. Liu

M. Toy
Verizon
L. Liu
Fujitsu
K. Makhijani
Futurewei
August 18, 2020

IS-IS Topology-Transparent Zone
draft-chen-isis-ttz-12.txt

Abstract

This document presents a topology-transparent zone in an area. A zone is a block/piece of an area, which comprises a group of routers and a number of circuits connecting them. It is abstracted as a virtual entity such as a single virtual node or zone edges mesh. Any router outside of the zone is not aware of the zone. The information about the circuits and routers inside the zone is not distributed to any router outside of the zone.

Status of This Memo

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

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on February 19, 2021.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Table of Contents

1. Introduction	3
1.1. Requirements Language	3
1.2. Terminology	3
2. Requirements	4
3. Zone Abstraction	4
4. Topology-Transparent Zone	5
4.1. Zone as a Single Node	5
4.1.1. An Example of Zone as a Single Node	5
4.1.2. Zone Leader Election	7
4.1.3. LS Generation for Zone as a Single Node	8
4.1.4. Adjacency Establishment and Termination	8
4.1.5. Computation of Routes	10
4.1.6. Extensions to Protocols	11
4.2. Zone as Edges Full Mesh	14
4.2.1. Extensions to IS-IS	14
4.3. Advertisement of LSs	15
4.3.1. Advertisement of LSs within Zone	15
4.3.2. Advertisement of LSs through Zone	16
5. Seamless Migration	16
5.1. Transfer Zone to a Single Node	16
5.2. Roll Back from Zone as a Single Node	16
6. Operations	19
7. Security Considerations	19
8. IANA Considerations	19
9. Contributors	20
10. Acknowledgement	20
11. References	20
11.1. Normative References	20
11.2. Informative References	21

Authors' Addresses	21
--------------------	----

1. Introduction

[ISO10589] describes two levels of areas, which are level 1 and level 2 areas in IS-IS. There are scalability issues in using areas as the number of routers in a network becomes larger and larger.

Through splitting the network into multiple areas, we may extend the network further. However, dividing a network from one area into multiple areas or from a number of existing areas to even more areas is a very challenging and time consuming task since it is involved in significant network architecture changes.

These issues can be resolved by using topology-transparent zone (TTZ), which abstracts a zone (i.e., a block/piece of an area) as a single virtual node or zone edges' mesh with minimum efforts and minimum service interruption. Note that a zone can be an area (i.e., the entire piece of an area).

This document presents a topology-transparent zone and describes extensions to IS-IS for supporting the topology-transparent zone.

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].

1.2. Terminology

LSP: A Link State Protocol Data Unit (PDU) in IS-IS.

LS: A Link State, which is short for LSP in IS-IS.

TTZ: A Topology-Transparent Zone.

Zone: A block or piece of an area. In a special case, a zone is an area (i.e., the entire piece of an area).

Zone External Node: A node outside of a zone.

Zone Internal Node: A node of a zone without any connection to a node outside of the zone.

Zone Edge/Border: A node of a zone connecting to a node outside of the zone.

Zone Node: A zone internal node or a zone edge/border node (i.e., a node of a zone).

Zone Link: A link connecting zone nodes (i.e., a link of a zone).

Zone Neighbor: A node outside of a zone that is a neighbor of a zone edge/border.

2. Requirements

Topology-Transparent Zone (TTZ) may be deployed for resolving some critical issues such as scalability in existing networks and future networks. The requirements for TTZ are listed as follows:

- o TTZ MUST be backward compatible. When a TTZ is deployed on a set of routers in a network, the routers outside of the TTZ in the network do not need to know or support TTZ.
- o TTZ MUST support at least one more levels of network hierarchies, in addition to the hierarchies supported by existing routing protocols.
- o Abstracting a zone as a virtual entity, which is a single virtual node or zone edges' mesh, SHOULD be smooth with minimum service interruption.
- o De-abstracting (or say rolling back) a virtual entity to a zone SHOULD be smooth with minimum service interruption.
- o Users SHOULD be able to easily set up an end to end service crossing TTZs.
- o The configuration for a TTZ in a network SHOULD be minimum.
- o The changes on the existing protocols for supporting TTZ SHOULD be minimum.

3. Zone Abstraction

A zone can be abstracted as a single virtual node or the zone edges' full mesh.

When a zone is abstracted as a single virtual node, this single node is connected to all the neighbors of the zone, and is in the same area as the neighbors.

When a zone is abstracted as its edges' full mesh, there is a full mesh connections among the edges and each edge is also connected to its neighbors outside of the zone.

4. Topology-Transparent Zone

A Topology-Transparent Zone (TTZ) comprises an Identifier (ID) and a piece/block of an area such as a Level 2 area in IS-IS. It is abstracted as a single virtual node or its edges' full mesh. TTZ and zone will be used exchangeably below.

4.1. Zone as a Single Node

After a zone is abstracted as a single virtual node having a virtual node ID, every node outside of the zone sees a number of links connected to this single node. Each of these links connects a zone neighbor. The link states inside the zone are not advertised to any node outside of the zone. The virtual node ID may be derived from the zone ID.

4.1.1. An Example of Zone as a Single Node

The figure below shows an example of an area containing a TTZ: TTZ 600.

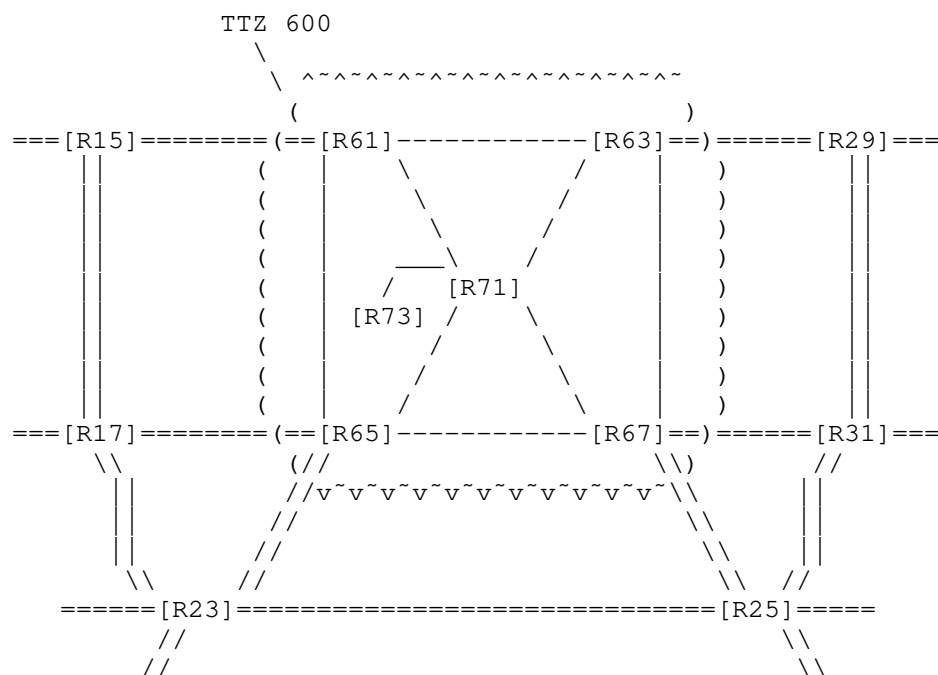


Figure 1: An Example of TTZ 600

The area comprises routers R15, R17, R23, R25, R29 and R31. It also contains TTZ 600, which comprises routers R61, R63, R65, R67, R71 and R73, and the circuits connecting them.

There are two types of routers in a TTZ: TTZ internal routers and TTZ edge/border routers. A TTZ internal router is a router inside the TTZ and its adjacent routers are inside the TTZ. A TTZ edge/border router is a router inside the TTZ and has at least one adjacent router that is outside of the TTZ.

The TTZ in the figure above comprises four TTZ edge/border routers R61, R63, R65 and R67. Each TTZ edge/border router is connected to at least one router outside of the TTZ. For instance, router R61 is a TTZ edge/border router since it is connected to router R15, which is outside of the TTZ.

In addition, the TTZ comprises two TTZ internal routers R71 and R73. A TTZ internal router is not connected to any router outside of the TTZ. For instance, router R71 is a TTZ internal router since it is not connected to any router outside of the TTZ. It is just connected to routers R61, R63, R65, R67 and R73 inside the TTZ.

A TTZ MUST hide the information inside the TTZ from the outside. It MUST NOT directly distribute any internal information about the TTZ to a router outside of the TTZ.

For instance, the TTZ in the figure above MUST NOT send the information about TTZ internal router R71 to any router outside of the TTZ in the routing domain; it MUST NOT send the information about the circuit between TTZ router R61 and R65 to any router outside of the TTZ.

From a router outside of the TTZ, a TTZ is seen as a single node (refer to the Figure below). For instance, router R15, which is outside of TTZ 600, sees TTZ 600 as a single node Rz, which has normal connections to R15, R29, R17 and R23, R25 and R31.

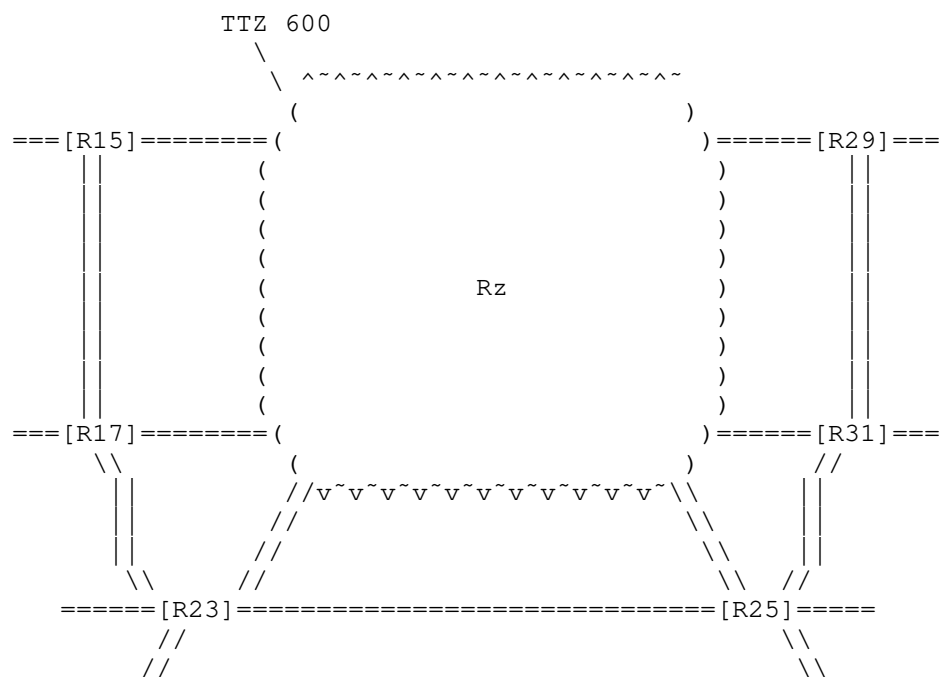


Figure 2: TTZ 600 as Single Node Rz

4.1.2. Zone Leader Election

A node in a zone is elected as a leader for the zone, which is the node with the highest priority (and the highest node ID when there are more than one nodes having the same highest priority) in the zone. The leader election mechanism described in

[I-D.ietf-lsr-dynamic-flooding] may be used to elect the leader for the zone.

4.1.3. LS Generation for Zone as a Single Node

The leader for the zone originates an LS (i.e., an LSP in IS-IS) for the zone as a single virtual node and sends it to its neighbors.

The LS comprises all the links connecting the zone neighbors. The LS ID is the ID of the virtual node for the zone. The Source ID or Advertising Node/Router ID is the ID of the virtual node.

In addition, the LS may contain the stub links for the routes such as the loopback addresses inside the zone to be accessed by zone external nodes (i.e., nodes outside of the zone).

4.1.4. Adjacency Establishment and Termination

A zone edge node, acting as a single virtual node for the zone, forms an adjacency with a node outside of the zone in a way described below.

Case 1 for a new adjacency (i.e., no adjacency exists between the edge and the node outside of the zone also called zone neighbor):

The edge node originates and sends the zone neighbor every protocol packet such as Hello, which contains the virtual node ID as Source ID.

When the edge node synchronizes its link state database (LSDB) with the zone neighbor, it sends the zone neighbor the information about all the link states except for the link states belonging to the zone that are hidden from any node outside of the zone.

At the end of the LSDB synchronization, the LS for the zone as the single virtual node is originated by the zone leader and distributed to the zone neighbor. This LS contains the links connecting all the zone neighbors, including this newly formed zone neighbor.

Case 2 for an existing adjacency (i.e., an adjacency already exists between the zone edge and the zone neighbor):

At first, the edge acting as virtual node creates a new adjacency between the virtual node for the zone and the zone external node in a normal way. It sends Hellos and other packets containing the virtual node ID as Source ID to the zone external node. The zone external node establishes the adjacency with the virtual in the normal way.

And then, the edge terminates the existing adjacency between the edge and the external node after the zone has been transferred to the virtual node. It stops sending Hellos for the adjacency to the zone external node. Without receiving Hellos from the edge node for a given time such as hold-timer interval, the zone external node removes the adjacency to the edge node. Even though this adjacency terminates, the edge node keeps the link to the external node in its LS.

In another option, the zone edge sends Hellos to the zone neighbor with additional information, including a flag T-bit set to one and a TLV with the virtual node ID. This information requests the zone neighbor to transfer the existing adjacency to the new adjacency smoothly through working together with the zone edge in following steps.

Zone Edge	Zone Neighbor
(Transfer Zone to Virtual Node)	
Hello(T=1, Virtual ID)	
	-----> OK for Transfer Adjacency
	Hello(T=1, Virtual ID)
Remote Ready for Transfer	<-----
	Hello(Source=Virtual ID)
Start Transfer	-----> Transfer to New Adjacency
	Hello
Transfer to New Adjacency	<-----
	. . .

Step 1: When "Transfer Zone to Virtual Node" is triggered, the zone edge sends the zone neighbor a Hello containing additional information T=1 and Virtual node ID.

Step 2: After receiving the Hello with T=1 and virtual node ID from the zone edge, the zone neighbor sends the zone edge a Hello with T=1 and virtual node ID, which means ok for transfer to the new adjacency.

Step 3: The edge sends the zone neighbor a Hello containing the virtual node ID as Source ID after receiving the Hello with T=1 and virtual node ID from the zone neighbor, which starts to transfer to the new adjacency.

Step 4: The zone neighbor changes the existing adjacency to the new adjacency after receiving the Hello containing the virtual node ID as Source ID from the zone edge; and sends the zone edge a Hello

without the additional information, which means that it transferred to the new adjacency.

Step 5: The zone edge changes the existing adjacency to the new adjacency after receiving the Hello without the additional information from the zone neighbor; and continues to send the zone neighbor a Hello containing the virtual node ID as Source ID. At this point, the old adjacency is transferred to the new one.

For the zone neighbor, changing the existing adjacency to the new one includes:

- o Changing the existing adjacency ID from the edge node ID to the virtual node ID through either removing the existing adjacency and adding a new adjacency with the virtual node ID or just changing the existing adjacency ID from the edge node ID to the virtual node ID,
- o Removing the link to the zone edge node from its LS and adding a new link to the virtual node (or just changing the link to the edge node to the link to the virtual node in its LS), and
- o Continuing sending the zone edge Hellos without additional information.

For the zone edge, changing the existing adjacency to the new one includes:

- o Keeping the link to the zone neighbor in its LS, and
- o Continuing sending the zone neighbor Hellos containing the virtual node ID as Source ID.

4.1.5. Computation of Routes

After a zone edge migrates to zone as a virtual node, it computes the routes (i.e., shortest paths to the destinations) in the zone using the zone topology (i.e., the topology of the zone without the virtual node).

For the routes outside of the zone, it computes them using the zone topology, the topology outside of the zone without the virtual node and the connections between each zone edge and its zone neighbor.

After a zone internal node migrates to zone as a virtual node, it computes the routes using the zone topology, the topology outside of the zone without the virtual node and the connections between each zone edge and its zone neighbor.

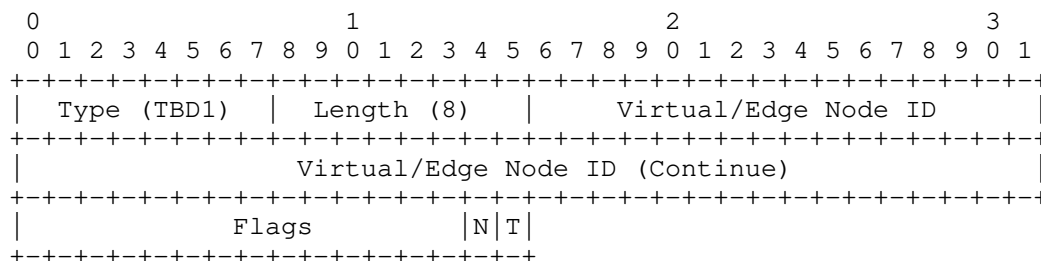
4.1.6. Extensions to Protocols

The following TLVs are defined in IS-IS.

- o Adjacent Node ID TLV: containing an adjacent node ID, to which an adjacency is transferred or rolled back. In case of transfer, the TLV contains the virtual node ID; in case of roll back, the TLV contains the edge node ID.
- o Zone TLV: containing a zone ID, a flags field and optional sub-TLVs.

4.1.6.1. Adjacent Node ID TLV

The format of Adjacent Node ID TLV is illustrated below.



Type (1 byte): To be assigned by IANA.

Length (1 byte): Its value is 8.

Virtual/Edge Node ID (6 bytes): An adjacent node ID, to which an adjacency is transferred or rolled back.

Flags field (16 bits): two new flag bits are defined as follows:

- o T-bit: Short for Transfer Adjacency bit. The T-bit set to one indicates a request for transferring to a new 'virtual' adjacency from the existing adjacency and the new adjacency is identified by the virtual node ID (or say abstract node ID).
- o N-bit: Short for Roll Back to Normal Adjacency bit. The N-bit set to one indicates a request for rolling back to a Normal adjacency from the existing 'virtual' adjacency and the normal adjacency is identified by the edge node ID.

4.1.6.2. Zone TLV

The format of IS-IS Zone TLV is illustrated below. It may be added into an LSP or a Hello PDU for a zone node. When a node in a zone receives a CLI command triggering zone information distribution for migration, it updates its LSP by adding an IS-IS Zone TLV with T set to 1. When a node in a zone receives a CLI command activating migration zone to an abstracted entity, it sets M to 1 in the Zone TLV in its LSP.

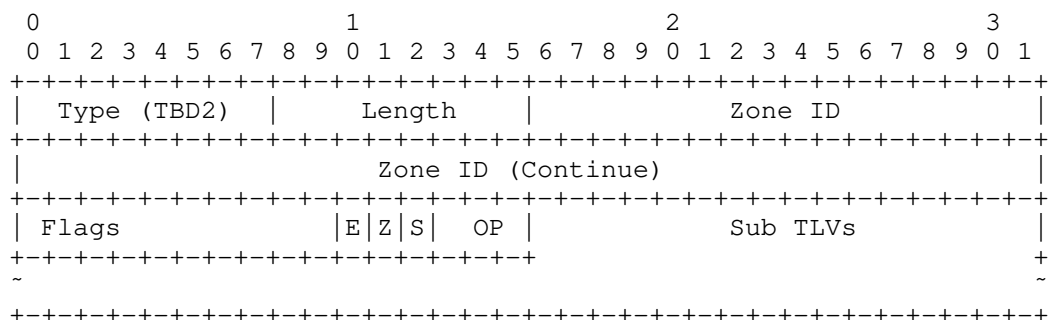


Figure 3: IS-IS Zone TLV

Type (1 byte): To be assigned by IANA.

Length (1 byte): Its value is variable.

Zone ID (6 bytes): It is the identifier (ID) of a zone.

Flags field (16 bits): Three flag bits E, Z and S, and OP of 3 bits are defined.

E = 1: Indicating a node is a zone edge node

Z = 1: Indicating a node has migrated to Zone as a virtual entity

S = 1: Indicating the virtual entity is a Single virtual node

When a zone node originates an LS containing a zone TLV, it MUST set flag E to 1 if it is a zone edge node and to 0 if it is a zone-internal node. It MUST set flag Z to 1 after it has migrated to zone as a virtual entity and to 0 before it migrates zone to the virtual entity or after it rolls back from zone as a virtual entity. When the entity abstracted from a zone is a Single virtual node, flag S MUST be set to 1.

OP Value	Meaning (Operation)
0x001 (T):	Advertising Zone Topology Information for Migration
0x010 (M):	Migrating Zone to a Virtual Entity
0x011 (N):	Advertising Normal Topology Information for Rollback
0x100 (R):	Rolling Back from the Virtual Entity

The value of OP indicates one of the four operations above. When any of the other values is received, it is ignored.

When a node in a zone receives a CLI command triggering zone information distribution for migration, it updates its LSP by adding an IS-IS Zone TLV with T set to 1. When a node in a zone receives a CLI command activating migration zone to a virtual entity, it sets M to 1 in the Zone TLV in its LSP.

Two new sub-TLVs are defined, which may be added into an IS-IS Zone TLV in an LSP. One is Zone IS Neighbor sub-TLV, or Zone ISN sub-TLV for short. The other is Zone ES Neighbor sub-TLV, or Zone ESN sub-TLV for short. A Zone ISN sub-TLV contains the information about a number of IS neighbors in the zone connected to a zone edge router. It has the format below.

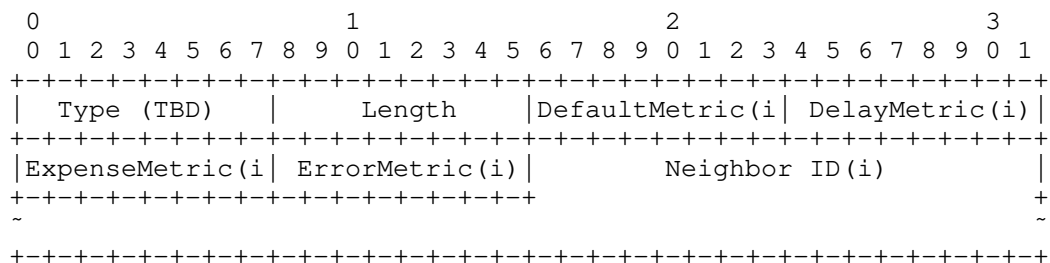


Figure 4: Zone ISN Sub TLV

A Zone ISN Sub TLV has 1 byte of Type, 1 byte of Length of $n \cdot (\text{IDLength} + 4)$, which is followed by n tuples of Default Metric, Delay Metric, Expense Metric, Error Metric and Neighbor ID.

A Zone ESN sub-TLV contains the information about a number of ES neighbors in the zone connected to a zone edge node. It has the format below.

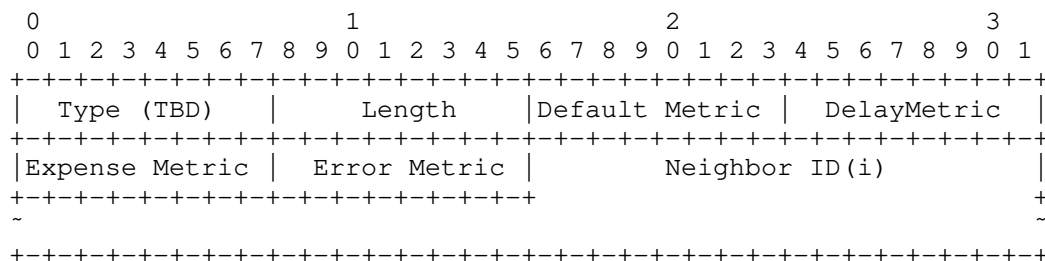


Figure 5: Zone ESN Sub TLV

4.2. Zone as Edges Full Mesh

OSPF Topology-Transparent Zone [RFC8099] describes the zone as edges' full mesh and the extensions to OSPF for supporting zone as edges' full mesh. Based on these extensions, IS-IS is extended by a few new TLVs or Sub-TLVs.

4.2.1. Extensions to IS-IS

4.2.1.1. Updating LSPs for Zone

A zone internal node adds an IS-IS Zone TLV into its LSP after it receives an LSP containing an IS-IS Zone TLV with T = 1 or a CLI command triggering zone information distribution for migration. The TLV has a zone ID set to the ID of the zone and E bit in Flags set to 0 indicating zone internal node. The node floods its LSP to its neighbors in the zone.

When a node inside the zone receives an LSP containing an IS-IS Zone TLV from a neighboring node in the zone, it stores the LSP and floods the LSP to the other neighboring nodes in the zone.

For every zone edge node, it updates its LSP in three steps and floods the LSP to all its neighbors.

At first, the zone edge node adds an IS-IS Zone TLV into its LSP after it receives an LSP containing an IS-IS Zone TLV with T = 1 or a CLI command triggering zone information distribution for migration. The TLV has a zone ID set to the ID of the zone, E bit in Flags set to 1 indicating zone edge node and a Zone ISN Sub TLV. The Sub TLV contains the information about the zone IS neighbors connected to the zone edge node. In addition, the TLV may has a Zone ESN Sub TLV comprising the information about the zone end systems connected to the zone edge node.

Secondly, it adds each of the other zone edge nodes as an IS neighbor into the Intermediate System Neighbors TLV in the LSP after it receives an LSP containing an IS-IS Zone TLV with $M = 1$ or a CLI command activating migration zone to an abstracted entity. The metric to the neighbor is the metric of the shortest path to the edge node within the zone.

In addition, it adds a Prefix Neighbors TLV into its LSP. The TLV contains a number of address prefixes in the zone to be reachable from outside of the zone.

And then it removes the IS neighbors corresponding to the IS neighbors in the Zone TLV (i.e., in the Zone ISN sub TLV) from Intermediate System Neighbors TLV in the LSP, and the ES neighbors corresponding to the ES neighbors in the Zone TLV (i.e., in the Zone ESN sub TLV) from End System Neighbors TLV in the LSP. This SHOULD be done after it receives the LSPs for virtualizing zone from the other zone edges for a given time.

4.3. Advertisement of LSs

LSs can be divided into a couple of classes according to their Advertisements. The first class of LSs is advertised within a zone. The second is advertised through a zone.

4.3.1. Advertisement of LSs within Zone

Any LS about a link state in a zone is advertised only within the zone. It is not advertised to any router outside of the zone. For example, a router LS generated for a zone internal router is advertised only within the zone.

Any network LS generated for a broadcast network in a zone is advertised only within the zone. It is not advertised outside of the zone.

After migrating to zone as a single virtual node or edges' full mesh, every zone edge MUST NOT advertise any LS belonging to the zone or any information in a LS belonging to the zone to any node outside of the zone. The zone edge determines whether an LS is about a zone internal link state by checking if the advertising router of the LS is a zone internal router.

For any zone LS originated by a node within the zone, every zone edge node MUST NOT advertise it to any node outside of the zone.

4.3.2. Advertisement of LSs through Zone

Any LS about a link state outside of a zone received by a zone edge is advertised using the zone as transit. For example, when a zone edge node receives an LS from a node outside of the zone, it floods the LS to its neighbors both inside and outside of the zone. This LS may be any LS such as a router LSA that is advertised within an OSPF area.

The nodes in the zone continue to flood the LS. When another zone edge receives the LS, it floods the LS to its neighbors both inside and outside of the zone.

5. Seamless Migration

This section presents the seamless migration between a zone and its single virtual node. The seamless migration between a zone and its edges' full mesh for IS-IS is similar to that described in OSPF Topology-Transparent Zone [RFC8099] for OSPF.

5.1. Transfer Zone to a Single Node

After transfer a Zone to a Single Virtual Node is triggered, the zone is abstracted as a single virtual node in two steps:

Step 1: Every zone edge node works together with each of its zone neighbor nodes to create a new adjacency between the virtual node and the neighbor node in the way described in Section 4.1.4 for Adjacency Establishment and Termination procedure for case 2. After creating the adjacency, each of the zone neighbor nodes update its LS by adding the adjacency/link into its LS.

Step 2: The zone leader originates an LS for the virtual node after receiving the updated LSes originated by all the zone neighbor nodes, where the updated LSes contain all the zone neighbors.

Step 3: After receiving the LS for the virtual node, every zone edge does not send any LS inside the zone to any zone neighbors. It advertises its LS without any links inside the zone to the nodes outside of the zone and terminates its adjacency to each of its zone neighbors in the way described in Section 4.1.4 for Adjacency Establishment and Termination procedure for case 2.

5.2. Roll Back from Zone as a Single Node

After roll back from Zone as a Single Virtual Node is triggered, rolling back is done in following steps:

Step 1: Every zone edge creates an adjacency to each of its zone neighbors in a normal way.

Step 2: After all the adjacencies between the zone edges and the zone neighbors are created, the zone leader updates the LS for the virtual node by changing every link metric to the maximum metric in the LS.

Step 3: Every zone edge sends its LS with the links inside the zone and all the LSes inside the zone to its zone neighbors. Every zone edge acting as the virtual node terminates the adjacency between the virtual node and each of its zone neighbors through stopping Hellos to the neighbors.

In another option, rolling back is done as follows:

Step 1: Using the procedure described in the following, every zone edge rolls back the existing virtual adjacency between the edge node acting as the virtual node and the zone neighbor node to a normal adjacency between the edge node and the neighbor.

Step 2: The zone leader may flush the LS for the virtual node. Every zone edge sends Hello and other packets to its zone neighbors, where the packets contain the edge node ID as Source ID.

The procedure below smoothly rolls back the existing virtual adjacency between the edge node acting as the virtual node and the zone neighbor node to a normal adjacency between the edge node and the neighbor node.

The edge node sends the neighbor node Hellos with additional information, including a flag N-bit set to one and a TLV with the edge node ID such as the Adjacent Node ID TLV with the edge node ID. This information requests the neighbor node to roll back the existing virtual adjacency to the normal adjacency smoothly through working together with the edge node.

The following steps will roll back the existing virtual adjacency to the normal one:

zone Edge (Roll Back to Normal Adjacency)		zone Neighbor
	Hello (N=1, Edge ID)	
	----->	OK to Roll Back to Normal Adjacency
	Hello (N=1, Edge ID)	
Remote Ready for Rolling Back	<-----	
	Hello (Source=Edge ID)	
Start Roll Back	----->	Roll Back to Normal Adjacency
	Hello	
Roll Back to Normal Adjacency	<-----	
	. . .	

Step 1: When "Roll Back from Zone as a Single Node" is triggered, the edge node sends the neighbor node a Hello with the additional information N=1 and Edge ID as normal adjacency ID in order to roll back to the normal adjacency from the virtual adjacency.

Step 2: After receiving the Hello with the additional information from the edge node, the neighbor node sends the edge node a Hello with the additional information (i.e., N=1 and Edge ID as normal adjacency ID), which means ok for rolling back to the normal adjacency.

Step 3: The edge sends the neighbor a Hello containing the edge node ID as Source ID after receiving the Hello with the additional information from the neighbor, which starts to roll back to the normal adjacency.

Step 4: The neighbor node changes the existing adjacency to the normal adjacency after receiving the Hello containing the edge node ID as Source ID from the edge node; and sends the edge node a Hello without the additional information, which means that it rolled back to the normal adjacency.

Step 5: The edge node changes the existing adjacency to the normal adjacency after receiving the Hello without the additional information from the neighbor node; and continues to send the neighbor Hello containing the edge node ID as Source ID. At this point, the virtual adjacency is rolled back to the normal adjacency.

For the neighbor node, changing the existing virtual adjacency to the normal one includes:

- o Changing the existing adjacency ID from the virtual node ID to the edge node ID through either removing the existing adjacency and adding a new adjacency with the edge node ID or just changing the existing adjacency ID from the virtual node ID to the edge node ID,
- o Removing the link to the virtual node from its LS and adding a new link to the edge node (or just changing the link to the virtual node to the link to the edge node in its LS), and
- o Continuing sending the edge node Hellos without additional information.

For the edge node, changing the existing virtual adjacency to the normal one includes:

- o Sending its LS to the neighbor, and
- o Continuing sending the neighbor node Hellos containing the edge node ID as Source ID without additional information.

6. Operations

The Operations on TTZ described in OSPF Topology-Transparent Zone [RFC8099] are for Zone as Edges Full Mesh in OSPF. They can be used for Zone as Edges Full Mesh in IS-IS. They can also be used for Zone as a Single Virtual Node in IS-IS.

7. Security Considerations

The mechanism described in this document does not raise any new security issues for the IS-IS protocols.

8. IANA Considerations

Under the registry name "IS-IS TLV Codepoints", IANA is requested to assign new registry types for Adjacent Node ID, Zone ID and Zone Options as follows:

TLV Type	TLV Name	reference
26(suggested)	Adjacent Node ID	This document
27(suggested)	Zone	This document

9. Contributors

Alvaro Retana
Futurewei
Raleigh, NC
USA

Email: alvaro.retana@futurewei.com

10. Acknowledgement

The authors would like to thank Acee Lindem, Abhay Roy, Christian Hopps, Dean Cheng, Russ White, Tony Przygienda, Wenhui Lu, Lin Han, Kiran Makhiyani, Padmadevi Pillay Esnault, and Yang Yu for their valuable comments on TTZ.

11. References

11.1. Normative References

- [I-D.ietf-lsr-dynamic-flooding]
Li, T., Psenak, P., Ginsberg, L., Chen, H., Przygienda, T., Cooper, D., Jalil, L., Dontula, S., and G. Mishra, "Dynamic Flooding on Dense Graphs", draft-ietf-lsr-dynamic-flooding-07 (work in progress), June 2020.
- [I-D.ietf-spring-segment-routing]
Filsfils, C., Previdi, S., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", draft-ietf-spring-segment-routing-15 (work in progress), January 2018.
- [ISO10589]
International Organization for Standardization, "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)", ISO/IEC 10589:2002, 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, <<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>>.

- [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>>.
- [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>>.
- [RFC7142] Shand, M. and L. Ginsberg, "Reclassification of RFC 1142 to Historic", RFC 7142, DOI 10.17487/RFC7142, February 2014, <<https://www.rfc-editor.org/info/rfc7142>>.
- [RFC8099] Chen, H., Li, R., Retana, A., Yang, Y., and Z. Liu, "OSPF Topology-Transparent Zone", RFC 8099, DOI 10.17487/RFC8099, February 2017, <<https://www.rfc-editor.org/info/rfc8099>>.

11.2. Informative References

- [Clos] Clos, C., "A Study of Non-Blocking Switching Networks", The Bell System Technical Journal Vol. 32(2), DOI 10.1002/j.1538-7305.1953.tb01433.x, March 1953, <<http://dx.doi.org/10.1002/j.1538-7305.1953.tb01433.x>>.
- [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>>.

Authors' Addresses

Huaimo Chen
Futurewei
Boston, MA
USA

Email: huaimo.chen@futurewei.com

Richard Li
Futurewei
2330 Central expressway
Santa Clara, CA
USA

Email: richard.li@futurewei.com

Yi Yang
IBM
Cary, NC
United States of America

Email: yyietf@gmail.com

Anil Kumar S N
RtBrick
Bangalore
India

Email: anil.ietf@gmail.com

Yanhe Fan
Casa Systems
USA

Email: yfan@casa-systems.com

Ning So
Plano, TX 75082
USA

Email: ningso01@gmail.com

Vic Liu
USA

Email: liu.cmri@gmail.com

Mehmet Toy
Verizon
USA

Email: mehmet.toy@verizon.com

Lei Liu
Fujitsu
USA

Email: liulei.kddi@gmail.com

Kiran Makhiyani
Futurewei
USA

Email: kiranm@futurewei.com

LSR Working Group
Internet-Draft
Intended status: Standards Track
Expires: 3 August 2022

J. Dong
Z. Hu
Z. Li
Huawei Technologies
X. Tang
R. Pang
China Unicom
L. JooHeon
LG U+
S. Bryant
Futurewei Technologies
30 January 2022

IGP Extensions for Scalable Segment Routing based Enhanced VPN
draft-dong-lsr-sr-enhanced-vpn-07

Abstract

Enhanced VPN (VPN+) aims to provide enhanced VPN services to support some application's needs of enhanced isolation and stringent performance requirements. VPN+ requires integration between the overlay VPN connectivity and the characteristics provided by the underlay network. A Virtual Transport Network (VTN) is a virtual underlay network which has a customized network topology and a set of network resources allocated from the physical network. A VTN could be used to support one or a group of VPN+ services.

This document specifies the IGP mechanisms with necessary extensions to advertise the associated topology and resource attributes for scalable Segment Routing (SR) based VTNs. Each VTN can have a customized topology and a set of network resources allocated from the physical network. Multiple VTNs may shared the same topology, and multiple VTNs may share the same set of network resources on some network segments. A group of resource-aware SIDs are allocated for each VTN. This allows flexible combination of the network topology and network resource attributes to build a relatively large number of VTNs with a small number of logical topologies. The proposed mechanism is applicable to both Segment Routing with MPLS data plane (SR-MPLS) and segment routing with IPv6 data plane (SRv6). This document also describes the mechanisms of using dedicated VTN-ID in the data plane instead of the per-VTN resource-aware SIDs to further reduce the control plane overhead.

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

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

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

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

This Internet-Draft will expire on 3 August 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
2. VTN Definition Advertisement	4
3. Advertisement of VTN Topology Attribute	6
3.1. MTR based Topology Advertisement	6
3.2. Flex-Algo based Topology Advertisement	7
4. Advertisement of VTN Resource Attribute	8
4.1. Option 1: L2 Bundle based Approach	8
4.2. Option 2: Per-VTN Link TE Attributes	10
5. Advertisement of VTN specific Data Plane Identifiers	12
5.1. Advertisement of VTN-specific SR-MPLS SIDs	12
5.2. Advertisement of VTN-specific SRv6 Locators and SIDs	14

5.2.1. VTN-specific SRv6 Locators and End SIDs	14
5.2.2. VTN-specific SRv6 End.X SIDs	17
5.3. Advertisement of Dedicated Data Plane VTN IDs	17
6. Security Considerations	18
7. IANA Considerations	18
8. Contributors	19
9. Acknowledgments	19
10. References	19
10.1. Normative References	19
10.2. Informative References	21
Authors' Addresses	21

1. Introduction

Enhanced VPN (VPN+) is an enhancement to VPN services to support the needs of new applications, particularly the applications that are associated with 5G services. These applications require enhanced isolation and have more stringent performance requirements than that can be provided with traditional overlay VPNs. These properties require integration between the underlay and the overlay networks. [I-D.ietf-teas-enhanced-vpn] specifies the framework of enhanced VPN and describes the candidate component technologies in different network planes and layers. An enhanced VPN can be used for 5G network slicing, and will also be of use in more generic scenarios.

To meet the requirement of different enhanced VPN services, a number of virtual underlay networks need to be created, each with a customized network topology and a set of network resources allocated from the physical network to meet the requirement of one or a group of VPN+ services. Such a virtual underlay network is called Virtual Transport Network (VTN) in [I-D.ietf-teas-enhanced-vpn].

[I-D.ietf-spring-resource-aware-segments] introduces resource-aware segments by associating existing type of SIDs with network resource attributes (e.g. bandwidth, processing or storage resources). These resource-aware SIDs retain their original functionality, with the additional semantics of identifying the set of network resources available for the packet processing action. [I-D.ietf-spring-sr-for-enhanced-vpn] describes the use of resource-aware segments to build SR based VTNs. To allow the network controller and network nodes to perform VTN-specific explicit path computation and/or shortest path computation, the group of resource-aware SIDs allocated by network nodes to each VTN and the associated topology and resource attributes need to be distributed using the control plane.

[I-D.dong-teas-nrp-scalability] analyzes the scalability requirements and the control plane and data plane scalability considerations of enhanced VPN, more specifically, the scalability of the VTNs. In order to support a relatively large number of VTNs in the network, one proposed approach is to separate the topology and resource attributes of the VTN in control plane, so that the advertisement and processing of each type of attribute could be decoupled. Multiple VTNs may share the same topology, and multiple VTNs may share the same set of network resources on some network segments, while the difference in either the topology or resource attributes makes them different VTNs. This allows flexible combination of network topology and network resource attributes to build a large number of VTNs with a relatively small number of logical topologies.

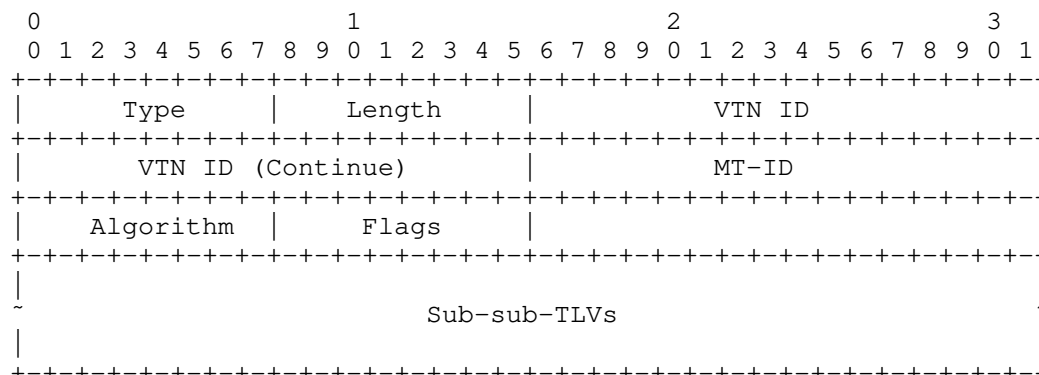
This document specifies the IGP control plane mechanisms with necessary extensions for scalable SR based VTNs. The proposed mechanism is applicable to both segment routing with MPLS data plane (SR-MPLS) and segment routing with IPv6 data plane (SRv6). This document also describes the mechanisms of using dedicated VTN-ID in the data plane instead of the per-VTN resource-aware SIDs to further reduce the control plane overhead.

In general this approach applies to both IS-IS and OSPF, while the specific protocol extensions and encodings are different. In the current version of this document, the required IS-IS extensions are described. The required OSPF extensions will be described in a future version or in a separate document.

2. VTN Definition Advertisement

According to [I-D.ietf-teas-enhanced-vpn], a VTN is associated with a customized network topology and a set of dedicated or shared network resources. Thus a VTN can be defined as the combination of a set of network attributes, which include the topology attribute and other attributes, such as the network resources. IS-IS Virtual Transport Network Definition (VTND) sub-TLV is used to advertise the definition of a VTN. It is a sub-TLV of the IS-IS Router-Capability TLV 242 as defined in [RFC7981].

The format of IS-IS VTND sub-TLV is as below:



Where:

- * Type: TBD
- * Length: The length of the value field of the sub-TLV. It is variable dependent on the included sub-TLVs.
- * VTN ID: A global significant 32-bit identifier which is used to identify a VTN.
- * MT-ID: 16-bit field which indicates the multi-topology identifier as defined in [RFC5120]. The first 4-bit are set to zero.
- * Algorithm: 8-bit identifier which indicates the algorithm which applies to this VTN. It can be either a normal algorithm [RFC8402] or a Flexible Algorithm [I-D.ietf-lsr-flex-algo].
- * Flags: 8-bit flags. Currently all the flags are reserved for future use. They SHOULD be set to zero on transmission and MUST be ignored on receipt.
- * Sub-sub-TLVs: optional sub-sub-TLVs to specify the additional attributes of a VTN. Currently no sub-sub-TLV is defined in this document.

The VTND Sub-TLV MAY be advertised in an LSP of any number. A node MUST NOT advertise more than one VTND Sub-TLV for a given VTN ID.

3. Advertisement of VTN Topology Attribute

This section describes the mechanisms used to advertise the topology attribute associated with SR based VTNs. Basically the topology of a VTN can be determined by the MT-ID and/or the algorithm ID included in the VTN definition. In practice, it could be described using two optional approaches.

The first approach is to use Multi-Topology Routing (MTR) [RFC4915] [RFC5120] with the segment routing extensions to advertise the topology associated with the SR based VTNs. Different algorithms MAY be used to further specify the computation algorithm or the metric type used for path computation within the topology. Multiple VTNs can be associated with the same <topology, algorithm>, and the IGP computation with the <topology, algorithm> tuple can be shared by these VTNs.

The second approach is to use Flex- Algo [I-D.ietf-lsr-flex-algo] to describe the topological constraints of SR based VTNs on a shared network topology (e.g. the default topology). Multiple VTNs can be associated with the same Flex- Algo, and the IGP computation with this Flex- Algo can be shared by these VTNs.

3.1. MTR based Topology Advertisement

Multi-Topology Routing (MTR) has been defined in [RFC4915] and [RFC5120] to create different network topologies in one network. It also has the capability of specifying customized attributes for each topology. The traditional use cases of multi-topology are to maintain separate topologies for unicast and multicast services, or to create different topologies for IPv4 and IPv6 in a network. There are some limitations when MTR is used with native IP forwarding, the considerations about MT based IP forwarding are described in [RFC5120].

MTR can be used with SR-MPLS data plane. [RFC8667] specifies the IS-IS extensions to support SR-MPLS data plane, in which the Prefix-SID sub-TLVs can be carried in IS-IS TLV 235 (MT IP Reachability) and TLV 237 (MT IPv6 IP Reachability), and the Adj-SID sub-TLVs can be carried in IS-IS TLV 222 (MT-ISN) and TLV 223 (MT IS Neighbor Attribute).

MTR can also be used with SRv6 data plane.

[I-D.ietf-lsr-isis-srv6-extensions] specifies the IS-IS extensions to support SRv6 data plane, in which the MT-ID is carried in the SRv6 Locator TLV. The SRv6 End SIDs are carried as sub-TLVs in the SRv6 Locator TLV, and inherit the topology/algorithm from the parent locator. The SRv6 End.X SIDs are carried as sub-TLVs in the IS-IS TLV 222 (MT-ISN) and TLV 223 (MT IS Neighbor Attribute), and inherit the topology/algorithm from the parent locator.

These IGP extensions for SR-MPLS and SRv6 can be used to advertise and build the topology for a group of SR based VTNs.

An algorithm ID MAY be used to further specify the computation algorithm or the metric type used for path computation within the topology.

3.2. Flex-Algo based Topology Advertisement

[I-D.ietf-lsr-flex-algo] specifies the mechanisms to provide distributed computation of constraint-based paths, and how the SR-MPLS prefix-SIDs and SRv6 locators can be used to steer packets along the constraint-based paths.

The Flex-Algo Definition (FAD) can be used to describe the topological constraints for path computation on a network topology. According to the network nodes' participation of a Flex-Algo, and the rules of including or excluding specific Administrative Groups (colors) and the Shared Risk Link Groups (SRLGs), the topology of a VTN can be determined using the associated Flex-Algo on a particular topology (e.g. the default topology).

With the mechanisms defined in[RFC8667] [I-D.ietf-lsr-flex-algo], prefix-SID advertisement can be associated with a <topology, algorithm> tuple, in which the algorithm can be a Flex-Algo. This allows network nodes to use the prefix-SID to steer traffic along distributed computed paths according to the identified Flex-Algo in the topology.

[I-D.ietf-lsr-isis-srv6-extensions] specifies the IS-IS extensions to support SRv6 data plane, in which the SRv6 locators advertisement can be associated with a specific topology and a specific algorithm, which can be a Flex-Algo. With the mechanism defined in [I-D.ietf-lsr-flex-algo], The SRv6 locator can be used to steer traffic along distributed computed paths according to the identified Flex-Algo in the topology. In addition, topology/algorithm specific SRv6 End SID and End.X SID can be used to enforce traffic over the LFA computed backup path.

Multiple Flex-Algos MAY be defined to describe the topological constraints on a shared network topology (e.g. the default topology).

4. Advertisement of VTN Resource Attribute

This section specifies the mechanisms to advertise the network resource attributes associated with the VTNs. The mechanism of advertising the link resources and attributes associated with VTNs is described. The mechanism of advertising node resources and attributes associated with VTNs are for further study. Two optional approaches are described in the following sub-sections: the first option is the L2 Bundle [RFC8668] based approach, the second option is to extend IGP to advertise per-VTN link TE attributes.

4.1. Option 1: L2 Bundle based Approach

On a Layer-3 interface, each VTN can be allocated with a subset of link resources (e.g. bandwidth). A subset of link resources may be dedicated to a VTN, or may be shared by a group of VTNs. Each subset of link resource can be represented as a virtual layer-2 member link under the Layer-3 interface, and the Layer-3 interface is considered as a virtual Layer-2 bundle. The Layer-3 interface may also be a physical Layer 2 link bundle, in this case a subset of link resources allocated to a VTN may be provided by one of the physical Layer-2 member links.

[RFC8668] describes the IS-IS extensions to advertise the link attributes of the Layer 2 member links which comprise a Layer 3 interface. Such mechanism can be extended to advertise the attributes of each physical or virtual member links, and its associated VTNs.

A new flag "E" (Exclusive) is defined in the flag field of the Parent L3 Neighbor Descriptor in the L2 Bundle Member Attributes TLV (25).

```

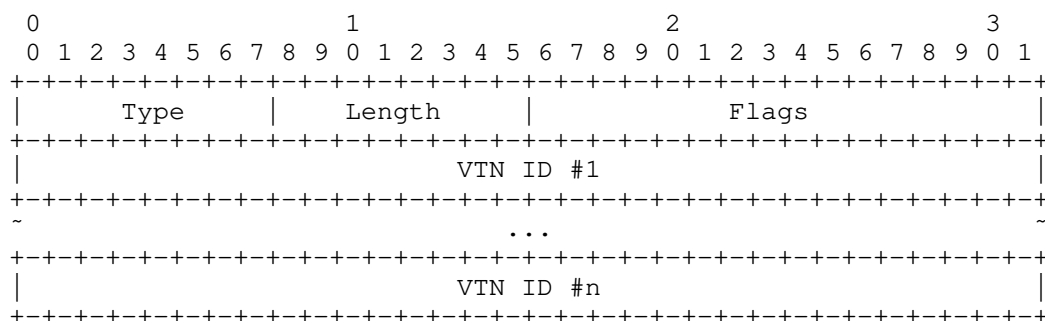
0 1 2 3 4 5 6 7
+--+--+--+--+--+--+
|P|E|          |
+--+--+--+--+--+--+

```

E flag: When the E flag is set, it indicates each member link under the Parent L3 link are used exclusively for one or a specific group of VTNs, and load sharing among the member links is not allowed. When the E flag is clear, it indicates load balancing and sharing among the member links are allowed.

A new VTN-IDs sub-TLV is carried under the L2 Bundle Attribute Descriptors to describe the mapping relationship between the VTNs and the virtual or physical member links. As one or more VTNs may use the same set of link resource on a specific network segment, these VTN IDs will be advertised under the same virtual or physical member link.

The format of the VTN-IDs Sub-TLV is as below:



Where:

- * Type: TBD
- * Length: The length of the value field of the sub-TLV. It is variable dependent on the number of VTN IDs included.
- * Flags: 16 bit flags. All the bits are reserved for future use, which SHOULD be set to 0 on transmission and MUST be ignored on receipt.
- * VTN IDs: One or more 32-bit identifier to identify the VTNs this member link belongs to.

Each physical or virtual member link MAY be associated with a different group of VTNs. Thus each L2 Bundle Attribute Descriptor may carry the link local identifier and attributes of only one Layer 2 member link. Multiple L2 Bundle Attribute Descriptors will be used to carry the attributes and the associated VTN-IDs of all the Layer 2 member links.

The TE attributes of each virtual or physical member link, such as the bandwidth attributes and the SR SIDs, can be advertised using the mechanism as defined in [RFC8668].

4.2. Option 2: Per-VTN Link TE Attributes

A Layer-3 interface can participate in multiple VTNs, each of which is allocated with a subset of the forwarding resources of the interface. For each VTN, the associated resources can be described using per-VTN TE attributes. A new VTN-specific TE attribute sub-TLV is defined to advertise the link attributes associated with a VTN. This sub-TLV MAY be advertised as a sub-TLV of the following TLVs:

TLV-22 (Extended IS reachability) [RFC5305]

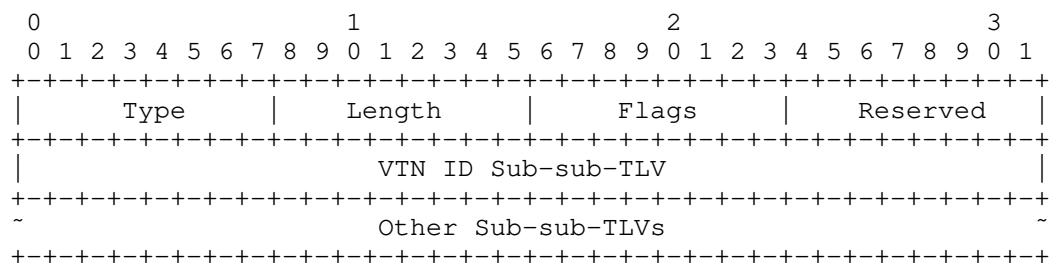
TLV-23 (IS Neighbor Attribute) [RFC5311]

TLV-141 (Inter-AS Reachability Information) [RFC5316]

TLV-222 (MT ISN) [RFC5120]

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

The format of the sub-TLV is shown as below:

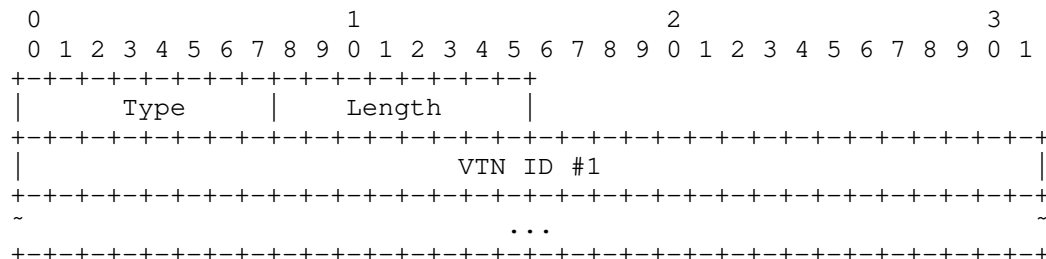


Where:

- * Type: TBD
- * Length: The length of the value field of the sub-TLV. It is variable dependent on the length of the Sub-sub-TLVs field.
- * Flags: 8-bit flags. All the 8 bits are reserved for future use, which SHOULD be set to 0 on transmission and MUST be ignored on receipt.
- * Reserved: 8-bit field reserved for future use, SHOULD be set to 0 on transmission and MUST be ignored on receipt.
- * VTN ID Sub-sub-TLV: contains one or more VTN IDs which is associated with the same group of TE attributes.

- * Other Sub-sub-TLVs: the TLVs which carry the TE attributes associated with the VTNs.

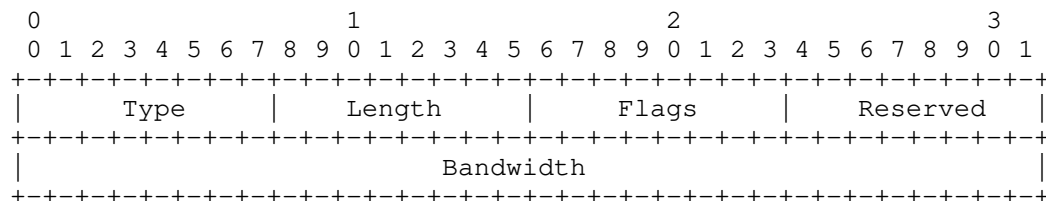
The format of the VTN ID sub-sub-TLV is shown as below:



Where:

- * Type: TBD
- * Length: The length of the value field of the sub-sub-TLV. It is the number of the VTN IDs in the TLV multiplied by 4.
- * VTN ID: A global significant 32-bit identifier which is used to identify a VTN.

One sub-sub-TLV "VTN bandwidth sub-sub-TLV" is defined in this document. Its format is shown as below:



Where:

- * Type: TBD
- * Length: The length of the value field of the sub-sub-TLV. It is set to 6.
- * Flags: 8-bit flags. All the 8 bits are reserved for future use, which SHOULD be set to 0 on transmission and MUST be ignored on receipt.

- * **Reserved:** 8-bit field reserved for future use, SHOULD be set to 0 on transmission and MUST be ignored on receipt.
- * **Bandwidth:** The bandwidth allocated to the VTN, encoded in 32 bits in IEEE floating point format.

The VTN-specific Bandwidth sub-sub-TLV is optional. This sub-sub-TLV SHOULD appear once at most in each VTN-specific TE attribute sub-TLV.

5. Advertisement of VTN specific Data Plane Identifiers

In order to steer packets to the VTN-specific paths which are computed taking the topology and network resources of the VTN as the constraints, some fields in the data packet needs to be used to infer or identify the VTN the packet belongs to. As multiple VTNs may share the same topology or Flex-Algo, the topology/Flex-Algo specific SR SIDs or Locators cannot be used to distinguish the packets which belong to different VTNs. Some additional data plane identifiers would be needed to identify the VTN a packet belongs to.

This section describes the mechanisms to advertise the VTN identifiers in different data plane encapsulations.

5.1. Advertisement of VTN-specific SR-MPLS SIDs

With SR-MPLS data plane, the VTN identification information can be implicitly carried in the VTN-specific SIDs. Each node SHOULD allocate a unique Prefix-SID for each VTN it participates in. On a Layer-3 interface, if each Layer 2 member link is associated with only one VTN, the adj-SIDs of the L2 member links could also identify the VTNs. If a member link is associated with multiple VTNs, VTN-specific adj-SIDs MAY need to be allocated to help the VTN-specific local protection.

A new VTN-specific prefix-SID sub-TLV is defined to advertise the prefix-SID and its associated VTN. This sub-TLV MAY be advertised as a sub-TLV of the following TLVs:

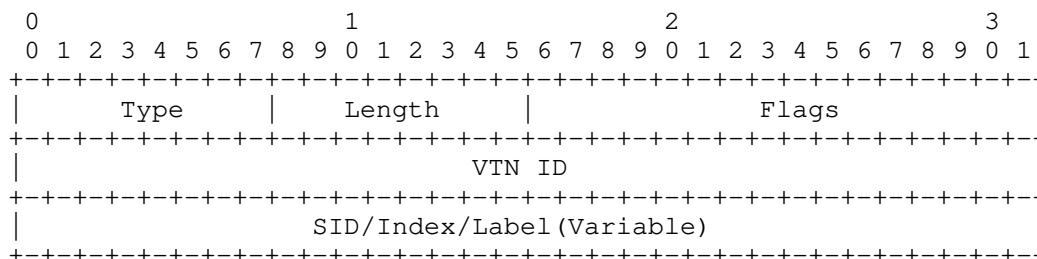
TLV-135 (Extended IPv4 Reachability) defined in [RFC5305].

TLV-235 (MT IP Reachability) defined in [RFC5120].

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

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

The format of the sub-TLV is shown as below:



Where:

- * Type: TBD
- * Length: The length of the value field of the sub-TLV. It is variable dependent on the length of the SID/Index/Label field.
- * Flags: 16-bit flags. The high-order 8 bits are the same as in the Prefix-SID sub-TLV defined in [RFC8667]. The lower-order 8 bits are reserved for future use, which SHOULD be set to 0 on transmission and MUST be ignored on receipt.
- * VTN ID: A 32-bit identifier to identify the VTN this prefix-SID associates with.
- * SID/Index/Label: The same as defined in [RFC8667].

One or more of VTN-specific Prefix-SID sub-TLVs MAY be carried in the Multi-topology IP Reachability TLVs (TLV 235 or TLV 237), the MT-ID of the TLV SHOULD be the same as the MT-ID in the definition of these VTNs.

A new VTN-specific Adj-SID sub-TLV is defined to advertise the adj-SID and its associated VTN. This sub-TLV may be advertised as a sub-TLV of the following TLVs:

TLV-22 (Extended IS reachability) [RFC5305]

TLV-23 (IS Neighbor Attribute) [RFC5311]

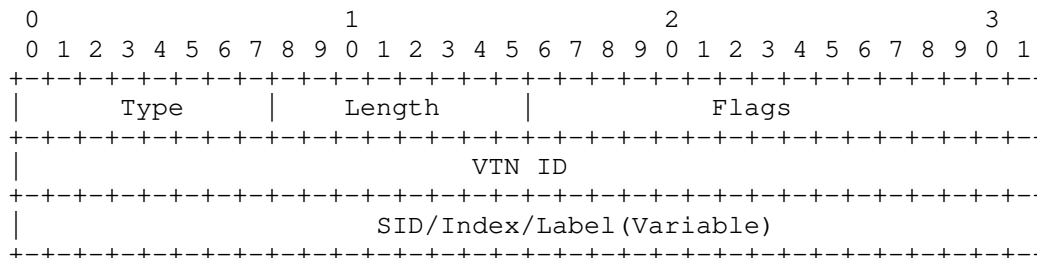
TLV-25 (L2 Bundle Member Attributes) [RFC8668]

TLV-141 (Inter-AS Reachability Information) [RFC5316]

TLV-222 (MT ISN) [RFC5120]

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

The format of the sub-TLV is shown as below:



Where:

- * Type: TBD
- * Length: The length of the value field of the sub-TLV. It is variable dependent on the length of the SID/Index/Label field.
- * Flags: 16-bit flags. The high-order 8 bits are the same as in the Adj-SID sub-TLV defined in [RFC8667]. The lower-order 8 bits are reserved for future use, which SHOULD be set to 0 on transmission and MUST be ignored on receipt.
- * VTN ID: A 32-bit global identifier to identify the VTN this Adj-SID associates with.
- * SID/Index/Label: The same as defined in [RFC8667].

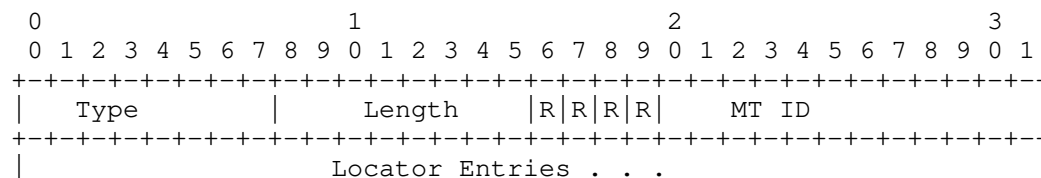
One or more VTN-specific Adj-SID sub-TLV MAY be carried in the Multi-topology ISN or Multi-topology IS Attribute TLVs (TLV 222 or TLV 223), the MT-ID of the TLV SHOULD be the same as the MT-ID in the definition of these VTNs.

5.2. Advertisement of VTN-specific SRv6 Locators and SIDs

5.2.1. VTN-specific SRv6 Locators and End SIDs

With SRv6 data plane, the VTN identification information can be implicitly or explicitly carried in the SRv6 Locator of the corresponding VTN, this is to ensure that all network nodes (including both the end nodes and the transit nodes) can identify the VTN to which a packet belongs to. Network nodes SHOULD allocate VTN-specific Locators for each VTN it participates in. The VTN-specific Locators are used as the covering prefix of VTN-specific SRv6 End SIDs, End.X SIDs and other types of SIDs.

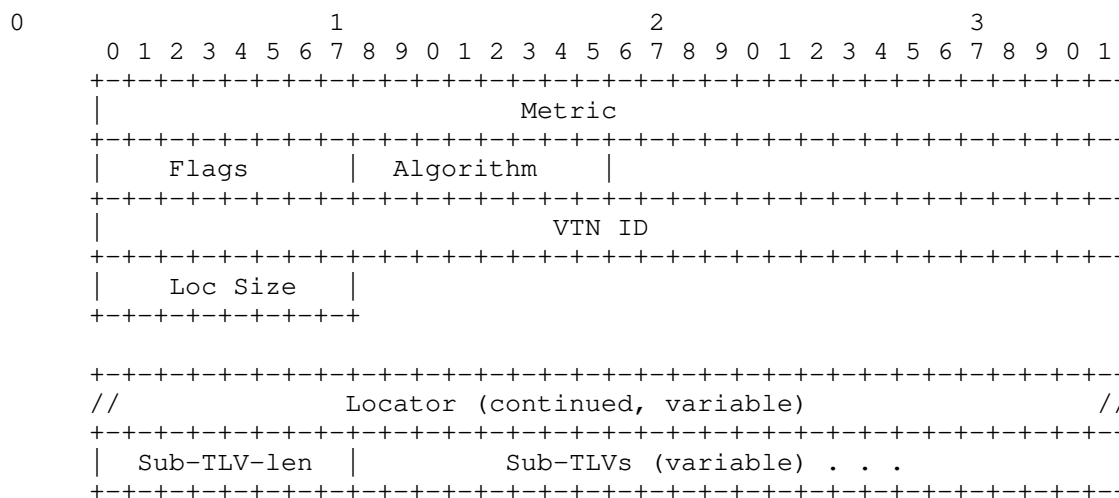
In one possible approach, each VTN-specific Locator is advertised in a separate TLV called "VTN specific SRv6 Locator TLV". Its format is shown as below:



Where:

- * Type: TBD
- * The semantics of the Length field, the R bits and the MT ID field are the same as those defined in [I-D.ietf-lsr-isis-srv6-extensions].

Followed by one or more locator entries of the form:



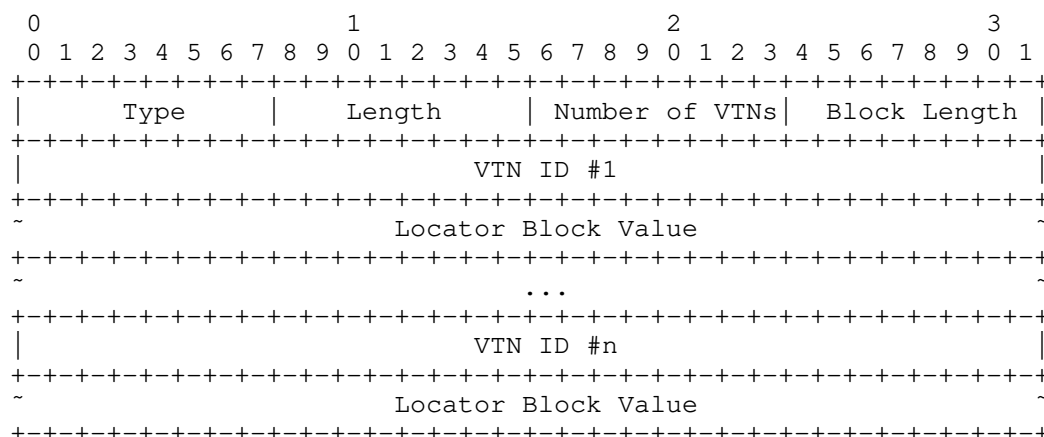
Where:

- * VTN ID: A 32-bit global identifier to identify the VTN this Locator associates with.
- * All the other fields are the same as those defined in [I-D.ietf-lsr-isis-srv6-extensions].

The VTN-specific SRv6 End SIDs are carried in the VTN-specific SRv6 Locator TLV, and inherits the topology, algorithm and VTN from the parent VTN-specific Locator.

In another possible approach, when a group of VTNs share the same topology/algorithm, the topology/algorithm specific Locator is the covering prefix of a group of VTN-specific Locators. Then the advertisement of VTN-specific locators can be optimized to reduce the amount of Locator TLVs advertised in the control plane.

A new VTN locator-block sub-TLV under the SRv6 Locator TLV is defined to advertise a set of sub-blocks which follows the topology/algorithm specific Locator. Each VTN locator-block value is assigned to one of the VTNs which share the same topology/algorithm.



Where:

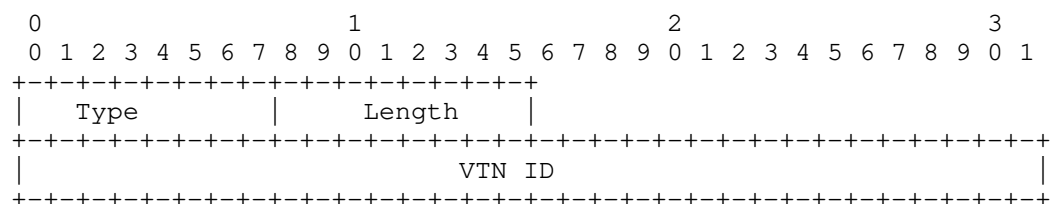
- * Type: TBD
- * Length: The length of the value field of the sub-TLV. It is variable dependent on the number of VTNs and the Block Length.
- * Number of VTNs: The number of VTNs which share the same topology/algorithm specific Locator as the covering prefix.
- * Block Length: The length of the VTN locator-block which follows the length of the topology/algorithm specific Locator.
- * VTN ID: A 32-bit global identifier to identify the VTN the locator-block is associates with.
- * Block Value: The value of the VTN locator-block for each VTN.

With the VTN locator-block sub-TLV, the VTN-specific Locator can be obtained by concatenating the topology/algorithm specific locator and the locator-block value advertised for the VTN.

The VTN-specific SRv6 End SIDs inherit the topology, algorithm and the VTN from the parent VTN-specific Locator.

5.2.2. VTN-specific SRv6 End.X SIDs

The SRv6 End.X SIDs are advertised as sub-TLVs of TLV 22, 23, 25, 141, 222, and 223. In order to distinguish the End.X SIDs which belong to different VTNs, a new "VTN ID sub-sub-TLV" is introduced under the SRv6 End.X SID sub-TLV and SRv6 LAN End.X SID sub-TLV defined in [I-D.ietf-lsr-isis-srv6-extensions]. Its format is shown as below:



Where:

- * Type: TBD.
- * Length: the length of the Value field of the TLV. It is set to 4.
- * VTN ID: A 32-bit global identifier to identify the VTN this End.X SID associates with.

5.3. Advertisement of Dedicated Data Plane VTN IDs

As the number of VTNs increases, with the mechanism described in [I-D.ietf-spring-sr-for-enhanced-vpn], the number of SR SIDs and SRv6 Locators allocated for different VTNs would also increase. In network scenarios where the number of SIDs or Locators becomes a concern, some data plane optimization may be needed to reduce the amount of SR SIDs and Locators allocated. As described in [I-D.dong-teas-nrp-scalability], one approach is to decouple the data plane identifiers used for topology based forwarding and the identifiers used for the VTN-specific processing. Thus a dedicated data plane VTN-ID could be encapsulated in the packet. One possible encapsulation of VTN-ID in IPv6 data plane is proposed in [I-D.dong-6man-enhanced-vpn-vtn-id]. One possible encapsulation of VTN-ID in MPLS data plane is proposed in

[I-D.li-mpls-enhanced-vpn-vtn-id].

In that case, the VTN-ID encapsulated in data plane can have the same value as the VTN-ID in control plane, so that the overhead of advertising the mapping between the control plane VTN-IDs and the corresponding data plane identifiers could be saved.

6. Security Considerations

This document introduces no additional security vulnerabilities to IS-IS.

The mechanism proposed in this document is subject to the same vulnerabilities as any other protocol that relies on IGPs.

7. IANA Considerations

IANA is requested to assign a new code point in the "sub-TLVs for TLV 242 registry".

Type: TBD1

Description: Virtual Transport Network Definition

IANA is requested to assign three new code points in the "sub-TLVs for TLVs 22, 23, 25, 141, 222, and 223 registry".

Type: TBD2

Description: Virtual Transport Network Identifiers

Type: TBD3

Description: VTN-specific TE attribute sub-TLV

Type: TBD4

Description: VTN-specific Adj-SID

IANA is requested to assign two new code points in the "Sub-TLVs for TLVs 27, 135, 235, 236 and 237 registry".

Type: TBD5

Description: VTN-specific Prefix-SID

Type: TBD6

Description: VTN locator-block

IANA is requested to assign a new code point in the "IS-IS TLV Codepoints registry".

Type: TBD7

Description: VTN-specific SRv6 Locator TLV

IANA is requested to assign a new code point in the "sub-sub-TLVs for SRv6 End SID and SRv6 End.X SID registry".

Type: TBD8

Description: VTN ID Sub-sub-TLV

8. Contributors

Hongjie Yang

Email: hongjie.yang@huawei.com

9. Acknowledgments

The authors would like to thank Mach Chen, Dean Cheng and Guoqi Xu for their review and discussion of this document.

10. References

10.1. Normative References

[I-D.ietf-lsr-flex-algo]

Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", Work in Progress, Internet-Draft, draft-ietf-lsr-flex-algo-18, 25 October 2021, <<https://www.ietf.org/archive/id/draft-ietf-lsr-flex-algo-18.txt>>.

[I-D.ietf-lsr-isis-srv6-extensions]

Psenak, P., Filsfils, C., Bashandy, A., Decraene, B., and Z. Hu, "IS-IS Extensions to Support Segment Routing over IPv6 Dataplane", Work in Progress, Internet-Draft, draft-ietf-lsr-isis-srv6-extensions-18, 20 October 2021, <<https://www.ietf.org/archive/id/draft-ietf-lsr-isis-srv6-extensions-18.txt>>.

[I-D.ietf-spring-resource-aware-segments]

Dong, J., Bryant, S., Miyasaka, T., Zhu, Y., Qin, F., Li, Z., and F. Clad, "Introducing Resource Awareness to SR Segments", Work in Progress, Internet-Draft, draft-ietf-spring-resource-aware-segments-03, 12 July 2021, <<https://www.ietf.org/archive/id/draft-ietf-spring-resource-aware-segments-03.txt>>.

- [I-D.ietf-spring-sr-for-enhanced-vpn]
Dong, J., Bryant, S., Miyasaka, T., Zhu, Y., Qin, F., Li, Z., and F. Clad, "Segment Routing based Virtual Transport Network (VTN) for Enhanced VPN", Work in Progress, Internet-Draft, draft-ietf-spring-sr-for-enhanced-vpn-01, 12 July 2021, <<https://www.ietf.org/archive/id/draft-ietf-spring-sr-for-enhanced-vpn-01.txt>>.
- [I-D.ietf-teas-enhanced-vpn]
Dong, J., Bryant, S., Li, Z., Miyasaka, T., and Y. Lee, "A Framework for Enhanced Virtual Private Network (VPN+) Services", Work in Progress, Internet-Draft, draft-ietf-teas-enhanced-vpn-09, 25 October 2021, <<https://www.ietf.org/archive/id/draft-ietf-teas-enhanced-vpn-09.txt>>.
- [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>>.
- [RFC4915] Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., and P. Pillay-Esnault, "Multi-Topology (MT) Routing in OSPF", RFC 4915, DOI 10.17487/RFC4915, June 2007, <<https://www.rfc-editor.org/info/rfc4915>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.

[RFC8668] Ginsberg, L., Ed., Bashandy, A., Filsfils, C., Nanduri, M., and E. Aries, "Advertising Layer 2 Bundle Member Link Attributes in IS-IS", RFC 8668, DOI 10.17487/RFC8668, December 2019, <<https://www.rfc-editor.org/info/rfc8668>>.

10.2. Informative References

[I-D.dong-6man-enhanced-vpn-vtn-id]
Dong, J., Li, Z., Xie, C., Ma, C., and G. Mishra,
"Carrying Virtual Transport Network (VTN) Identifier in
IPv6 Extension Header", Work in Progress, Internet-Draft,
draft-dong-6man-enhanced-vpn-vtn-id-06, 24 October 2021,
<<https://www.ietf.org/archive/id/draft-dong-6man-enhanced-vpn-vtn-id-06.txt>>.

[I-D.dong-teas-nrp-scalability]
Dong, J., Li, Z., Gong, L., Yang, G., Guichard, J. N.,
Mishra, G., and F. Qin, "Scalability Considerations for
Network Resource Partition", Work in Progress, Internet-
Draft, draft-dong-teas-nrp-scalability-00, 17 December
2021, <<https://www.ietf.org/archive/id/draft-dong-teas-nrp-scalability-00.txt>>.

[I-D.li-mpls-enhanced-vpn-vtn-id]
Li, Z. and J. Dong, "Carrying Virtual Transport Network
Identifier in MPLS Packet", Work in Progress, Internet-
Draft, draft-li-mpls-enhanced-vpn-vtn-id-01, 14 April
2021, <<https://www.ietf.org/archive/id/draft-li-mpls-enhanced-vpn-vtn-id-01.txt>>.

Authors' Addresses

Jie Dong
Huawei Technologies

Email: jie.dong@huawei.com

Zhibo Hu
Huawei Technologies

Email: huzhibo@huawei.com

Zhenbin Li
Huawei Technologies

Email: lizhenbin@huawei.com

Xiongyan Tang
China Unicom

Email: tangxy@chinaunicom.cn

Ran Pang
China Unicom

Email: pangran@chinaunicom.cn

Lee JooHeon
LG U+

Email: playgame@lguplus.co.kr

Stewart Bryant
Futurewei Technologies

Email: stewart.bryant@gmail.com

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: March 7, 2020

S. Dontula
ATT
T. Li
Arista Networks
September 4, 2019

YANG Data Model for Dynamic Flooding
draft-dontula-lsr-yang-dynamic-flooding-01

Abstract

This document defines YANG data models that can be used to configure and manage Dynamic Flooding for IS-IS and OSPF.

Status of This Memo

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

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

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

This Internet-Draft will expire on March 7, 2020.

Copyright Notice

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

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

Table of Contents

1. Requirements Language	2
2. Introduction	2
3. OSPF Data Model	2
4. IS-IS Data Model	9
5. Acknowledgements	18
6. Security Considerations	18
7. Normative References	18
Authors' Addresses	19

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. Introduction

This document defines a YANG [RFC7950] data model for IS-IS [ISO10589] and OSPF [RFC2328] routing protocols.

3. OSPF Data Model

```
module ietf-dynamic-flooding-ospf {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-dynamic-flooding-ospf";

  prefix ospf-df;

  import ietf-ospf {
    prefix "ospf";
    reference "RFC XXXX - YANG model for Open Shortest Path First (OSPF)";
    Please replace XXXX with published RFC number for draft-
    ietf-ospf-yang-21.";
  }
  import ietf-yang-types {
    prefix "yang";
    reference "RFC 6991: Common YANG Data Types";
  }

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

  organization
    "IETF LSR - Link State Routing Working Group";
```



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

  Editor: Srinath Dontula
    <mailto: sd947e@att.com>
  Author: Tony Li
    <mailto: tony.li@tony.li>
  Author: Peter Psenak
    <mialto: ppsenak@cisco.com>
  Author: Les Ginsberg
    <mailto: ginsberg@cisco.com>
  Author: Huaimo Chen
    <mailto: hcen@futurewei.com>
  Author: Tony Przygienda
    <mailto: prz@juniper.net>
  Author: Dave Cooper
    <mailto: dave.cooper@centurylink.com>
  Author: Luay Jalil
    <mailto: luay.jalil@verizon.com>";

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

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

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

revision 2019-08-16 {
  description
    "Initial draft version.";
  reference
    "RFC XXXX: A YANG data model for OSPF Dense topology";
}
```



```

    feature dynamic-flooding-ospf {
        description
            "support for OSPF dynamic flooding";
        reference " RFC XXXX - Dynamic Flooding on Dense Graphs";
    }

/*
*     identity ospfv2-dynamic-flooding-opaque-lsa {
*         base ospf:ospfv2-area-scope-opaque-lsa;
*         description
*             "OSPFv2 dynamic-flooding-opaque-lsa. RFC XXX- Replace XX
XX with published RFC number for draft-ietf-lsr-dynamic-flooding ";
*     }
*
*     identity ospfv3-dynamic-flooding-opaque-lsa {
*         base ospf:ospfv3-lsa-type;
*         description
*             "OSPFv3 dynamic-flooding-opaque-lsa. RFC XXX- Replace XX
XX with published RFC number for draft-ietf-lsr-dynamic-flooding ";
*     }
*/

    identity ospfv2-dynamic-flooding-opaque-lsa {
        base ospf:ospfv2-opaque-lsa-type;
        description "OSPFv2 Dynamic Flooding Opaque LSA - Type TBD";
    }

    identity ospfv3-dynamic-flooding-opaque-lsa {
        base ospf:ospfv3-lsa-type;
        description "OSPFv3 Dynamic Flooding LSA - Type TBD";
    }

    grouping ospf-area-leader-tlv {
        description "Area leader TLV for OSPFv2/OSPFv3. A TLV of RI LSA"
;
        leaf priority {
            type uint8;
            description "Router's priority for Area leader";
        }
        leaf Algorithm {
            type uint8;
            description "Routers Algorithm to calculate Flooding Top
ology.";
        }
    }

    grouping dynamic-flooding-tlv {
        description "Dynamic-flooding TLV for OSPFv2/OSPFv3. A TLV of RI
LSA";
        leaf-list algorithm {

```



```

        type uint8;
        description "List of Algorithm's supported by the advert
ising node";
    }
}

grouping ospfv2-area-router-id-tlv-entry {
    description "ospfv2 area router ID TLV enties";
    leaf connection-type {
        type uint8;
        description "Defines the connection type ";
    }
    leaf number-of-ids {
        type uint16;
        description "the number of ID present in this TLV entry"
;
    }
    leaf-list originating-router-id {
        type yang:dotted-quad;
        description "list of the originating-router-ids";
    }
}

grouping ospfv2-area-router-id-tlv {
    description "definition for OSPFv2 Dynamic flooding area router-
ID TLV";
    leaf start-index {
        type uint16;
        description "Starting index of the first router/designat
ed router ID";
    }
    leaf last-flag {
        type bits {
            bit last-router-designated-router-id {
                description "when set, this TLV is the l
ast Index in the full list of router IDs for the area";
                reference "RFC XXXX- Dynamic flooding on
Dense graphs; Replace XXXX with published RFC number for
draft-ietf-lsr-d
ynamic-flooding-03";
            }
        }
        description "last flag";
    }
    list ospfv2-router-id-tlv-entry {
        description "list of ospfv2 router-ID TLV entries";
        uses ospfv2-area-router-id-tlv-entry;
    }
}

grouping ospfv3-area-router-id-tlv-entry {
    description "ospfv3 area router ID TLV enties";
    leaf connection-type {
        type uint8;

```



```

        description "Defines the connection type ";
    }
    leaf number-of-ids {
        type uint16;
        description "the number of ID present in this TLV entry"
;
    }
    choice originating-router-id {
        description "list of the originating-router-ids";
        container router-id {
            when "derived-from(..connection-type,'router') "
{
                description "Only applies when connectio
n type is Router";
            }
            description "originating-router-id";
            leaf-list originating-router-id {
                type yang:dotted-quad;
                description "list of the originating-rou
ter-ids";
            }
        }
        container designated-router-id {
            when "derived-from(..connection-type,'designate
d-router') " {
                description "Only applies when connectio
n type is designated Router";
            }
            description "originating-router-id";
            list originating-router-id-list {
                description "originating-router-id";
                leaf originating-router-id{
                    type yang:dotted-quad;
                    description "originating-router-
ids";
                }
                leaf interface-id{
                    type yang:dotted-quad;
                    description "interface-ids";
                }
            }
        }
    }
}

grouping ospfv3-area-router-id-tlv {
    description "definition for OSPFv3 Dynamic flooding area router-
ID TLV";
    leaf start-index {
        type uint16;
        description "Starting index of the first router/designat
ed router ID";
    }
    leaf last-flag {
        type bits {
            bit last-router-designated-router-id {
                description "when set, this TLV is the 1
ast Index in the full list of router IDs for the area";
            }
        }
    }
}

```



```

        reference "RFC XXXX- Dynamic flooding on
Dense graphs; Replace XXXX with published RFC number for draft-ietf-lsr-dynamic
-flooding-03";
    }
    }
    description "dynamic flooding area router-ID flag defini
tions";
}
list ospfv3-router-id-tlv-entry {
    description "list of OSPFv3 router-ID TLV entries";
    uses ospfv3-area-router-id-tlv-entry;
}
}

grouping flooding-path-tlv {
    description "definition for OSPFv2/OSPFv3 Flooding Path TLV";
    leaf start-index {
        type uint16;
        description "Starting index of the first router/designat
ed router ID";
    }
    leaf-list indices {
        type uint16;
        description "index of the next router ID in the path";
    }
}

grouping flooding-request-bit{
    description "definition for adding Flooding request bit (FR-bit)
to LLS type-1 extended options. Need to be defined";
}

grouping link-attribute-bits-tlv{
    description "definition for link-attribute-bits-tlv, a sub-tlv f
or OSPFv2 Extended link TLV and a sub-tlv for OSPFv3 router-link TLV";
    leaf link-attributes-bits{
        type bits {
            bit LEEF {
                description "when set, conveys which edg
es are currently enabled in the flooding topology";
            }
        }
        description "link-attributes-bits";
    }
}

/*
* Dynamic flooding config augmentation to OSPF Module
*/
augment "/rt:routing/rt:control-plane-protocols/rt:control-plane-protoco
l/ospf:ospf/ospf:areas/ospf:area" {
    if-feature dynamic-flooding-ospf;
    description "Dynamic-flooding config model augmentation";
    container dynamic-flooding {
        description "to Enable/Disable dynamic flooding for this
specific OSPF area";
    }
}

```



```

        leaf enable {
            type boolean;
            description "Enable/Disable dynamic-flooding";
        }
    }

/*
 * Dynamic flooding state augmentation to OSPF2 module
 */

    augment "/rt:routing/rt:control-plane-protocols/rt:control-plane-protoco
l/ospf:ospf/ospf:areas/ospf:area/ospf:database/ospf:area-scope-lsa-type/ospf:are
a-scope-lsas/ospf:area-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2/ospf:body/
ospf:opaque" {
        if-feature dynamic-flooding-ospf;
        description "dynamic flooding TLVs augmentation";
        container area-leader-tlv {
            description "Area-leader-tlv";
            uses ospf-area-leader-tlv;
        }
        container dynamic-flooding-tlv {
            description "dynamic-flooding-tlv";
            uses dynamic-flooding-tlv;
        }
    }

    augment "/rt:routing/rt:control-plane-protocols/rt:control-plane-protoco
l/ospf:ospf/ospf:areas/ospf:area/ospf:database/ospf:area-scope-lsa-type/ospf:are
a-scope-lsas/ospf:area-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2/ospf:body"
    {
        if-feature dynamic-flooding-ospf;
        description "ospfv2-dynamic-flooding-opaque-lsa augmentation";
        container ospfv2-dynamic-flooding-opaque-lsa {

            description "OSPFv2 Dynamic flooding opaque LSA.";
            uses unknown-tlvs;
            container ospfv2-area-router-id-tlv {
                description "ospfv2-area-router-id-tlv";
                uses ospfv2-area-router-id-tlv;
            }
            container flooding-path-tlv {
                description "ospf flooding-path-tlv";
                uses flooding-path-tlv;
            }
        }
    }

    augment "/rt:routing/rt:control-plane-protocols/rt:control-plane-protoco
l/ospf:ospf/ospf:areas/ospf:area/ospf:database/ospf:area-scope-lsa-type/ospf:are
a-scope-lsas/ospf:area-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2/ospf:body/
ospf:opaque/ospf:extended-link-opaque/ospf:extended-link-tlv" {
        if-feature dynamic-flooding-ospf;
        description "ospf-link-attributes-bits-tlv augmentation";
        container ospf-link-attributes-bits-tlv {
            description "ospf link attributes bits tlv. RFC XXXX- Dy
namic flooding on Dense graphs; Replace XXXX with published RFC number for draft
-ietf-lsr-dynamic-flooding-03";
            uses link-attribute-bits-tlv;

```



```

    }
  }

/*
 * Dynamic flooding state augmentation to OSPF3 module
 */
    augment "/rt:routing/rt:control-plane-protocols/rt:control-plane-protoco
l/ospf:ospf/ospf:areas/ospf:area/ospf:database/ospf:area-scope-lsa-type/ospf:are
a-scope-lsas/ospf:area-scope-lsa/ospf:version/ospf:ospfv3/ospf:ospfv3/ospf:body/
ospf:router-information" {
        if-feature dynamic-flooding-ospf;
        description "dynamic flooding TLVs augmentation";
        container area-leader-tlv {
            description "Area-leader-tlv";
            uses ospf-area-leader-tlv;
        }
        container dynamic-flooding-tlv {
            description "dynamic-flooding-tlv";
            uses dynamic-flooding-tlv;
        }
    }

    augment "/rt:routing/rt:control-plane-protocols/rt:control-plane-protoco
l/ospf:ospf/ospf:areas/ospf:area/ospf:database/ospf:area-scope-lsa-type/ospf:are
a-scope-lsas/ospf:area-scope-lsa/ospf:version/ospf:ospfv3/ospf:ospfv3/ospf:body"
    {
        if-feature dynamic-flooding-ospf;
        description "ospfv3-dynamic-flooding-opaque-lsa augmentation";
        container ospfv3-dynamic-flooding-opaque-lsa {

            description "ospfv3-dynamic-flooding-opaque-lsa definiti
on";
            //
            uses unknown-tlvs;
            container ospfv3-area-router-id-tlv {
                description "ospfv3-area-router-id-tlv";
                uses ospfv3-area-router-id-tlv;
            }
            container flooding-path-tlv {
                description "ospf flooding-path-tlv";
                uses flooding-path-tlv;
            }
        }
    }
}

```

4. IS-IS Data Model

This data model augments the IS-IS YANG model [I-D.ietf-isis-yang-isis-cfg] with extensions to support Dynamic Flooding [I-D.ietf-lsr-dynamic-flooding].

```

module ietf-isis-dynflood {
    yang-version 1.1;

```



```
// IETF:
namespace "urn:ietf:params:xml:ns:yang:ietf-isis-dynflood";
// OpenConfig:
// namespace "http://openconfig.net/yang/openconfig-isis-dynflood";

prefix isis-dynflood;

// OpenConfig:
// import openconfig-network-instance {
//   prefix oc-netinst;
// }

// NB: This module is currently under development and is intended to augment
// the IETF IS-IS module (currently draft-ietf-isis-yang-isis-cfg-35).
// While that is in process, we are developing against the openconfig model, a
nd
// references to IETF paths are commented out.

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

import ietf-isis {
  prefix "isis";
  reference "https://tools.ietf.org/id/draft-ietf-isis-yang-isis-cfg-35.txt";
}

organization "IETF LSR Working Group";
contact "WG List: <mailto:lsr@ietf.org>";
description "
```

First draft of a YANG model for IS-IS dynamic flooding.

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

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

This version of this YANG module is part of RFC XXXX
(<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.

Authors:

Srinath Dontula
 <mailto:sd947e@att.com>

Tony Li
 <mailto:tony.li@tony.li>;

```

revision 2019-08-22 {
  description "Initial revision";
  reference "RFC XXXX";
}

// OpenConfig:
// 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";
// }

feature area-leader {
  description "Election of a leader for the area.";
}

feature dynamic-flooding {
  if-feature area-leader;
  description "Compute a reduced flooding topology.";
}

grouping area-leader-global-cfg {
  description "Enable area leader capability";
  leaf value {
    type boolean;
    default false;
    description "Enable area leader capability";
  }
}

```



```
}

grouping area-leader-priority-cfg {
  description "Set the area leader priority";
  leaf value {
    type uint8;
    default 200;
    description "Area leader priority";
  }
}

grouping area-leader-algorithm-cfg {
  description "Select the flooding topology computation algorithm";
  leaf value {
    type uint8;
    default 0;
    description "Dynamic flooding algorithm selection";
  }
}

grouping area-leader-parameters {
  description "Area leader configuration parameters";
  container area-leader {
    description "Area leader configuration parameters";
    container config {
      description "Area leader configuration";
      container enable {
        if-feature area-leader;
        description "Area leader global enable configuration";
        uses area-leader-global-cfg;
        container level-1 {
          description "level-1 specific configuration";
          uses area-leader-global-cfg;
        }
        container level-2 {
          description "level-2 specific configuration";
          uses area-leader-global-cfg;
        }
      }
    }

    container priority {
      if-feature area-leader;
      description "Area leader priority configuration";
      uses area-leader-priority-cfg;
      container level-1 {
        description "level-1 specific configuration";
        uses area-leader-priority-cfg;
      }
    }
  }
}
```



```
        container level-2 {
            description "level-2 specific configuration";
            uses area-leader-priority-cfg;
        }
    }

    container algorithm {
        if-feature dynamic-flooding;
        description "Area leader algorithm configuration";
        uses area-leader-algorithm-cfg;
        container level-1 {
            description "level-1 specific configuration";
            uses area-leader-algorithm-cfg;
        }
        container level-2 {
            description "level-2 specific configuration";
            uses area-leader-algorithm-cfg;
        }
    }
}

grouping dynamic-flooding-global-cfg {
    description "Enable dynamic flooding capability";
    leaf value {
        type boolean;
        default false;
        description "Enable dynamic flooding capability";
    }
}

grouping dynamic-flooding-parameters {
    description "Dynamic flooding configuration parameters";
    container dynamic-flooding {
        description "Dynamic flooding configuration parameters";
        container config {
            description "Dynamic flooding configuration";
            container enable {
                if-feature dynamic-flooding;
                description "Dynamic flooding global enable configuration";
                uses dynamic-flooding-global-cfg;
                container level-1 {
                    description "level-1 specific configuration";
                    uses dynamic-flooding-global-cfg;
                }
                container level-2 {
                    description "level-2 specific configuration";
```



```
        uses dynamic-flooding-global-cfg;
    }
}
}
}

// IETF:
augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/isis:isis" {
// OpenConfig:
// augment "/oc-netinst:network-instances/oc-netinst:network-instance/oc-netin
st:protocols/oc-netinst:protocol/oc-netinst:isis/oc-netinst:global" {
    description "Area leader and dynamic flooding configuration additions";
    uses area-leader-parameters;
    uses dynamic-flooding-parameters;
}

grouping dynamic-flooding-paths {
    description "List of paths in the topology";
    list paths {
        config false;
        description "A list of paths";
        leaf-list path {
            type uint16;
            description "A list of node indicies";
        }
    }
}

grouping dynamic-flooding-topology {
    description "List of paths in the topology";
    list paths {
        config false;
        description "A list of paths";
        leaf-list path {
            type string;
            description "A list of node names";
        }
    }
}

grouping dynamic-flooding-interfaces {
    description "List of flooding topology interfaces";
    list interfaces {
        config false;
        description "List of interfaces and their temporary status";
        leaf interface {
            type string;
        }
    }
}
```



```
        description "Interface name";
    }
    leaf temporary {
        type boolean;
        description "Set for partition repair or new adjacencies";
    }
}

grouping area-leader-state {
    description "Area leader state information";
    container area-leader-state {
        description "Area leader election result";
        leaf area-leader {
            type string;
            description "Hostname of the area leader";
        }
    }
}

grouping dynamic-flooding-state {
    description "State information for dynamic flooding";
    container area-node-ids {
        if-feature dynamic-flooding;
        description "Area node ids and indices";
        container area-node-ids {
            description "List of nodes and indices";
            list node-index {
                key index;
                description "List of nodes and their indices";
                leaf index {
                    type uint16;
                    description "Index for the node";
                }
                leaf node-id {
                    // OpenConfig:
                    // type extended-system-id;
                    // IETF:
                    type isis:extended-system-id;
                    description "Node id for the node";
                }
            }
        }
    }
}

container dynamic-flooding-paths {
    if-feature dynamic-flooding;
    description "Flooding topology with explicit paths";
    uses dynamic-flooding-paths;
}
```



```
    }
    container dynamic-flooding-topology {
        if-feature dynamic-flooding;
        description "Flooding topology with system names";
        uses dynamic-flooding-topology;
    }
    container dynamic-flooding-interfaces {
        if-feature dynamic-flooding;
        description "Interfaces on the flooding topology";
        uses dynamic-flooding-interfaces;
    }
}

// IETF:
augment "/rt:routing/rt:control-plane-protocols/"
    + "rt:control-plane-protocol/isis:isis" {
// OpenConfig:
// augment "/oc-netinst:network-instances/oc-netinst:network-instance/oc-netinst:protocols/oc-netinst:protocol/oc-netinst:isis/oc-netinst:levels/oc-netinst:level/oc-netinst:state" {
    description "Area leader and dynamic flooding state additions";
    uses area-leader-state;
    uses dynamic-flooding-state;
}

grouping subtlv27-area-leader {
    description "Router capability subTLV for area leader capability";
    container area-leader {
        description "Area leader subTLV for router capabilities";
        leaf priority {
            type uint8;
            description "Area leader priority";
        }
        leaf algorithm {
            type uint8;
            description "Algorithm index for computing the flooding topology";
        }
    }
}

grouping subtlv28-dynamic-flooding {
    description "Dynamic flooding capability subTLV";
    container dynamic-flooding {
        description "Dynamic flooding capability subTLV";
        leaf-list algorithms {
            type uint8;
            description "Supported algorithm indices for distributed mode";
        }
    }
}
```



```
// OpenConfig:
// augment "/oc-netinst:network-instances/oc-netinst:network-instance/oc-netinst:protocols/oc-netinst:protocol/oc-netinst:isis/oc-netinst:levels/oc-netinst:level/oc-netinst:link-state-database/oc-netinst:lsp/oc-netinst:tlvs/oc-netinst:tlv/oc-netinst:router-capabilities/oc-netinst:capability/oc-netinst:subtlvs/oc-netinst:subtlv" {
// IETF:
augment "/rt:routing/rt:control-plane-protocols"
  +"/rt:control-plane-protocol/isis:isis"
  +"/isis:database/isis:levels/isis:lsp/isis:router-capabilities/isis:router-capability" {
  description "Additional router capability subTLVs for dynamic flooding";
  uses subtlv27-area-leader;
  uses subtlv28-dynamic-flooding;
}

grouping tlv17-area-node-ids {
  description "TLV 17: Area Node IDs";
  container area-node-ids {
    description "Dynamic flooding node id assignment TLV";
    leaf starting-index {
      type uint8;
      description "Starting index for the node ids in this TLV.";
    }
    leaf flags {
      type bits {
        bit last {
          position 0;
          description "Set if this is the highest set of indices in the area."
        }
      }
    }
    description "Flags field in the TLV.";
  }
  leaf-list nodes {
    // OpenConfig:
    // type extended-system-id;
    // IETF:
    type isis:extended-system-id;
    description "Nodes being assigned indices";
  }
}

grouping tlv18-flooding-path {
  description "Flooding topology path TLV";
  container flooding-path {
    description "Dynamic flooding path TLV";
    leaf-list nodes {
      type uint16;
      description "Nodes in the path";
    }
  }
}
```



```
// OpenConfig:
// augment "/oc-netinst:network-instances/oc-netinst:network-instance/oc-netinst:protocols/oc-netinst:protocol/oc-netinst:isis/oc-netinst:levels/oc-netinst:level/oc-netinst:link-state-database/oc-netinst:lsp/oc-netinst:tlvs/oc-netinst:tlv" {
// IETF:
augment "/rt:routing/rt:control-plane-protocols"
  +"/rt:control-plane-protocol/isis:isis/isis:database/isis:levels/isis:lsp" {
    description "Additional TLVs for dynamic flooding";
    uses tlv17-area-node-ids;
    uses tlv18-flooding-path;
  }
}
```

5. Acknowledgements

The authors would like to thank Derek Yeung, Nidhi Bhaskar, and Kavitha Prasad for their comments and assistance.

6. Security Considerations

This document introduces no new security issues.

7. 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-35 (work in progress), March 2019.
- [I-D.ietf-lsr-dynamic-flooding]
Li, T., Psenak, P., Ginsberg, L., Chen, H., Przygienda, T., Cooper, D., Jalil, L., and S. Dontula, "Dynamic Flooding on Dense Graphs", draft-ietf-lsr-dynamic-flooding-03 (work in progress), June 2019.
- [ISO10589]
International Organization for Standardization, "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)", ISO/IEC 10589:2002, 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>>.
- [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <<https://www.rfc-editor.org/info/rfc2328>>.

[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>>.

Authors' Addresses

Srinath Dontula
ATT
200 S. Laurel Ave
Middletown, New Jersey 07748
United States of America

Email: sd947e@att.com

Tony Li
Arista Networks
5453 Great America Parkway
Santa Clara, California 95054
United States of America

Email: tony.li@tony.li

Networking Working Group
Internet-Draft
Intended status: Informational
Expires: January 9, 2022

L. Ginsberg
P. Psenak
M. Karasek
A. Lindem
Cisco Systems
T. Przygienda
Juniper
July 8, 2021

IS-IS Flooding Scale Considerations
draft-ginsberg-lsr-isis-flooding-scale-05

Abstract

Link State PDU flooding rates in use are much slower than what modern networks can support. The use of IS-IS at larger scale requires faster flooding rates to achieve desired convergence goals. This document discusses issues associated with increasing flooding rates and some recommended practices which allow faster flooding rates to be used safely.

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

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

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

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

This Internet-Draft will expire on January 9, 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
2. Historical Behavior	3
3. Flooding Rate and Convergence	4
3.1. Flow Control Considerations	5
3.2. Rate of LSP Acknowledgments	7
3.3. Bandwidth Utilization	7
3.4. Packet Prioritization on Receive	7
4. Minimizing LSP Generation	8
5. Redundant Flooding	10
6. Use of Jumbo Frames	10
7. Deployment Considerations	10
8. IANA Considerations	11
9. Security Considerations	11
10. Acknowledgements	11
11. References	11
11.1. Normative References	11
11.2. Informative References	12
Authors' Addresses	12

1. Introduction

Link state IGPs such as Intermediate-System-to-Intermediate-System (IS-IS) depend upon having consistent Link State Databases (LSDB) on all Intermediate Systems (ISs) in the network in order to provide correct forwarding of data packets. When topology changes occur, new/updated Link State PDUs (LSPs) are propagated network-wide. The speed of propagation is a key contributor to convergence time.

Historically, flooding rates have been conservative - on the order of 10s of LSPs/second. This derives from guidance in the base specification [ISO10589] and early deployments when both CPU speeds

and interface speeds were much slower than they are today and the scale of an IS-IS area was smaller than it may be today.

As IS-IS is deployed in greater scale (larger number of nodes in an area and larger number of neighbors/node), the impact of the historic flooding rates becomes more significant. Consider the bringup or failure of a node with 1000 neighbors. This will result in a minimum of 1000 LSP updates. At a typical LSP flooding rate used in many deployments today (33 LSPs/second), it would take 30+ seconds simply to send the updated LSPs to a given neighbor. Depending on the diameter of the network, achieving a consistent LSDB on all nodes in the network could easily take a minute (or more).

Increasing LSP flooding rate therefore becomes an essential element of supporting greater network scale.

The remainder of this document discusses various aspects of protocol operation and how they are impacted by increased flooding rate. Where appropriate, best practices are defined which enhance an implementation's ability to support faster flooding rates.

2. Historical Behavior

The base specification for IS-IS [ISO10589] was first published in 1992 and updated in 2002. The update made no changes in regards to suggested timer values. Convergence targets at the time were on the order of seconds and the specified timer values reflect that. Here are some examples:

minimumLSPGenerationInterval - This is the minimum time interval between generation of Link State PDUs. A source Intermediate system shall wait at least this long before re-generating one of its own Link State PDUs.

The recommended value was 30 seconds.

minimumLSPTransmissionInterval - This is the amount of time an Intermediate system shall wait before further propagating another Link State PDU from the same source system.

The recommended value was 5 seconds.

partialSNPInterval - This is the amount of time between periodic action for transmission of Partial Sequence Number PDUs.

It shall be less than minimumLSPTransmission-Interval.

The recommend value was 2 seconds.

Most relevant to a discussion of LSP flooding rate is the recommended interval between the transmission of two different LSPs on a given interface.

For broadcast interfaces, [ISO10589] defined:

minimumBroadcastLSPTransmissionInterval - the minimum interval between PDU arrivals which can be processed by the slowest Intermediate System on the LAN.

The default value was defined as 33 milliseconds.

NOTE: It was permitted to send multiple LSPs "back-to-back" as a burst, but this was limited to 10 LSPs in a one second period.

Although this value was specific to LAN interfaces, this has commonly been applied by implementations to all interfaces though that was not the original intent of the base specification. In fact Section 12.1.2.4.3 states:

On point-to-point links the peak rate of arrival is limited only by the speed of the data link and the other traffic flowing on that link.

Although modern implementations have not strictly adhered to the 33 millisecond interval, it is commonplace for implementations to limit flooding rate to an order of magnitude similar to the 33 ms value.

In the past 20 years, significant work on achieving faster convergence - more specifically sub-second convergence - has resulted in implementations modifying a number of the above timers in order to support faster signaling of topology changes. For example, minimumLSPGenerationInterval has been modified to support millisecond intervals - often with a backoff algorithm applied to prevent LSP generation storms in the event of a series of rapid oscillations.

However, flooding rate has not been fundamentally altered.

3. Flooding Rate and Convergence

Convergence involves a number of sequential operations.

First the topology change needs to be detected. This is a local activity occurring only on the node or nodes directly connected to the topology change. The directly connected node(s) then must advertise the topology change by updating their LSPs and flooding the changed LSPs. Routers then must process the updated LSDB and

recalculate paths to affected destinations. The updated paths must then be installed in the forwarding plane.

Only when all of the steps are completed on all nodes in the network has the network completed convergence.

As the convergence requirement is consistency of LSDBs on all nodes in the network, it is fundamental to understand that the goal of flooding is to update the LSDB on all nodes in the network "as fast as possible". Controlling the rate of flooding per interface is done to address some practical limitations which include:

- o Fairness to other data and control traffic on the same interface
- o Limitations on the processing rate of incoming control traffic

However, intentionally using different flooding rates on different interfaces increases the possibility of longer periods of LSDB inconsistency, which, in turn, delays network wide convergence.

Many implementations provide knobs to control the rate of LSP flooding on a per interface basis. To the extent that this serves as a flow control mechanism, this may reduce the number of dropped LSPs during high activity bursts and thereby reduce the number of LSP retransmissions required. As LSP retransmission timers are typically long (multiple seconds), this may result in shorter convergence times than if the LSP burst was uncontrolled. But if the performance characteristics of routers in the network are such that some routers consistently accept and process fewer LSPs/second than other routers, convergence will be degraded. Tuning LSP transmission timers on a per interface basis will never provide optimal convergence. Consistent flooding rates should be used on all interfaces.

3.1. Flow Control Considerations

In large scale deployments where an increased flooding rate is being used, it becomes more likely that a burst of LSPs may temporarily overwhelm a receiver. Normal operation of the Update Process will recover from this, but it may well make sense to employ some form of flow control. This will not serve to optimize convergence, but it can serve to reduce the number of LSP retransmissions. As retransmissions are deliberately done at a slow rate, the result of flow control will be to provide a shorter recovery time from a transient condition which prevents a node from handling the targeted rate of LSP transmission. Sustained inability to handle LSP reception at the targeted flooding rate indicates that the network is provisioned in a way which does not support optimal convergence. Steps need to be taken to resolve this issue. Such steps could

include upgrading the routers that demonstrate this condition consistently, altering the configuration on the problematic routers or altering the position of the problematic routers in the network so as to reduce the overall load on those routers, or reducing the target maximum LSP transmission rate network-wide.

When flow control is necessary, it can be implemented in a straightforward manner based on knowledge of the current flooding rate and the current acknowledgement rate. Such an algorithm is a local matter and there is no requirement or intent to standardize an algorithm. There are a number of aspects which serve as guidelines which can be described.

A maximum target LSP transmission rate (LSPTxMax) SHOULD be configurable. This represents the fastest LSP transmission rate which will be attempted. This value SHOULD be applicable to all interfaces and SHOULD be consistent network wide.

When the current rate of LSP transmission (LSPTxRate) exceeds the capabilities of the receiver, the flow control algorithm needs to aggressively reduce the LSPTxRate within a few seconds. Slower responsiveness is likely to result in a large number of retransmissions which can introduce much larger delays in convergence.

NOTE: Even with modest increases in flooding speed (for example, a target LSPTxMax of 300 LSPs/second (10 times the typical rate supported today)), a topology change triggering 2100 new LSPs would only take 7 seconds to complete.

Dynamic adjustment of the rate of LSP transmission (LSPTxRate) upwards (i.e., faster) SHOULD be done less aggressively and only be done when the neighbor has demonstrated its ability to sustain the current LSPTxRate.

The flow control algorithm MUST NOT assume the receive capabilities of a neighbor are static, i.e., it MUST handle transient conditions which result in a slower or faster receive rate on the part of a neighbor.

The flow control algorithm needs to consider the expected delay time in receiving an acknowledgment. See Section 3.2. This may vary per neighbor.

3.2. Rate of LSP Acknowledgments

On point-to-point networks, PSNP PDUs provide acknowledgments for received LSPs. [ISO10589] suggests that some delay be used when sending PSNPs. This provides some optimization as multiple LSPs can be acknowledged in a single PSNP.

If faster LSP flooding is to be used safely, it is necessary that LSPs be acknowledged more promptly as well. This requires a reduction in the delay in sending PSNPs.

As PSNPs also consume link bandwidth and packet queue space and protocol processing time on receipt, the increased sending of PSNPs should be taken into account when considering the rate at which LSPs can be sent on an interface.

3.3. Bandwidth Utilization

Routing protocol traffic has to share bandwidth on a link with other control traffic and data traffic. During periods of instability, routing protocol traffic will increase, but it is still desirable that the maximum bandwidth consumption by routing protocol traffic be modest. This needs to be considered when setting IS-IS flooding rates.

If we assume a maximum size of 1492 bytes for an LSP, here are some rough estimates of bandwidth consumption at different flooding rates:

LSPs/second	100 Mb Link	1 Gb Link
100	1.2 %	0.1 %
500	6.1 %	0.6 %
1000	12.1 %	1.2 %

3.4. Packet Prioritization on Receive

There are three classes of PDUs sent by IS-IS:

- o Hellos
- o LSPs

- o Complete Sequence Number PDUs (CSNPs) and Partial Sequence Number PDUs (PSNPs)

Implementations today may prioritize the reception of Hellos over LSPs and SNPs in order to prevent a burst of LSP updates from triggering an adjacency timeout which in turn would require additional LSPs to be updated.

SNPs serve to acknowledge or trigger the transmission of specified LSPs. On a point-to-point link, PSNPs acknowledge the receipt of one or more LSPs. Because PSNPs (like all IS-IS PDUs) use TLVs in the body, it is possible to acknowledge multiple LSPs using a single PSNP. For this reason, [ISO10589] specifies a delay (partialSNPInterval) before sending a PSNP so that the number of PSNPs required to be sent is reduced. On receipt of a PSNP, the set of LSPs acknowledged by that PSNP can be marked so that they do not need to be retransmitted.

If a PSNP is dropped on reception, this has a significant impact as the set of LSPs advertised in the PSNP cannot be marked as acknowledged and this results in needless retransmissions which may further delay transmission of other LSPs which have yet to be transmitted. It may also make it more likely that a receiver becomes overwhelmed by LSP transmissions.

It is therefore recommended that implementations prioritize the receipt of SNPs over LSPs.

4. Minimizing LSP Generation

In IS-IS the unit of flooding is an LSP. Each router may generate a set of LSPs at each supported level. Each LSP in the set has an LSP number - which is a value from 0-N where N = 255 for the base protocol. (N has been extended to 65535 by [RFC7356].) Each LSP carries network information using defined Type/Length/Value (TLV) tuples. For example, some TLVs carry neighbor information and some TLVs carry reachable prefix information. [ISO10589] strongly recommends preserving the association of a given advertisement (such as a neighbor) with a specific LSP whenever possible. This minimizes the number of LSPs which need to be regenerated when a topology change occurs. This recommendation becomes even more important as the scale of the network increases.

Consider the following example;

Node A has 11 neighbors currently in the UP state and is advertising them in three LSPs with content as follows:

A.00-00 contains the following advertisements

- Neighbor 1
- Neighbor 2
- Neighbor 3
- Neighbor 4
- Neighbor 5

A.00-01 contains the following advertisements:

- Neighbor 6
- Neighbor 7
- Neighbor 8
- Neighbor 9
- Neighbor 10

A.00-02 contains the following advertisements

- Neighbor 11

Imagine that the adjacency to Neighbor 3 goes down. There are (at least) two ways that A could update its LSPs.

Method 1: Node A removes the neighbor advertisement for neighbor 3 from A.00-00 and sends an update for that LSP. LSPs 00-01 and 00-02 are unchanged and so do not have to be flooded.

Method 2: Node A attempts to reduce the number of LSPs currently active and updates the content as follows:

A.00-00 contains the following advertisements

- Neighbor 1
- Neighbor 2
- Neighbor 4
- Neighbor 5
- Neighbor 6

A.00-01 contains the following advertisements:

- Neighbor 7
- Neighbor 8
- Neighbor 9
- Neighbor 10
- Neighbor 11

A.00-02 becomes empty

Node A now has to flood all three LSPs. LSPs #0 and #1 are reflooded because their content has changed. LSP #2 is purged.

In a large scale network, the impact of using Method #2 becomes significant and introduces conditions where a much larger number of LSPs need to be flooded than is the case with Method #1.

In order to operate at scale, implementations need to follow the guidance in [ISO10589] and use Method #1 whenever possible.

5. Redundant Flooding

Default operation of the Update Process is to flood on all interfaces. In cases where a network is highly meshed, this can result in a significant amount of redundant flooding. Nodes will receive multiple copies of each updated LSP.

There are defined mechanisms which can greatly reduce the redundant flooding. These include:

- o Mesh Groups ([RFC2973])
- o Dynamic Flooding ([I-D.ietf-lsr-dynamic-flooding])

6. Use of Jumbo Frames

The maximum size of an LSP (LSPBufferSize) is a parameter that needs to be set consistently network wide. This is because IS-IS does not support fragmentation of its PDUs - so in order for network wide flooding of an LSP to be successful all routers must restrict their LSP size to a size which can be supported without fragmentation on all interfaces on which IS-IS operates.

In networks where all interfaces on which IS-IS operates support large frames, LSPBufferSize may be set to a larger value than the default (1492). This allows more routing information to be encoded in a single LSP, which means that fewer LSPs are generated by each node and therefore the number of LSPs which need to be flooded can be reduced in some scenarios (e.g., node or interface bringup).

7. Deployment Considerations

As noted earlier in this document, it is desired to have consistent flooding speeds on all nodes in the network. Today, this is roughly achieved to the extent that current implementations flood at rates which are on the order of what is discussed in [ISO10589] , i.e., 33 LSPs/second).

As the goal is to introduce an order of magnitude increase in the rate of flooding (e.g., 10 times the current flooding rate) a network which has a mixture of nodes which support the faster flooding speeds and nodes which do not is at greater risk of introducing longer periods of LSDB inconsistency in the network - which is likely to have a negative impact on convergence and increase the occurrence of traffic drops or looping.

It is recommended that all nodes in the network support increased flooding rates before enabling use of the increased flooding rates.

Note that as the Update process runs in the context of an area (or the L2 sub-domain), enablement can safely be done on a per area basis even when nodes in another area do not support the faster flooding rates.

8. IANA Considerations

This document requires no actions by IANA.

9. Security Considerations

Security concerns for IS-IS are addressed in [ISO10589, [RFC5304], and [RFC5310].

10. Acknowledgements

Thanks to Bruno Decraene for his careful review and insightful comments.

11. References

11.1. Normative References

- [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>>.
- [RFC2973] Balay, R., Katz, D., and J. Parker, "IS-IS Mesh Groups", RFC 2973, DOI 10.17487/RFC2973, October 2000, <<https://www.rfc-editor.org/info/rfc2973>>.
- [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>>.
- [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>>.

11.2. Informative References

- [I-D.ietf-lsr-dynamic-flooding]
Li, T., Psenak, P., Ginsberg, L., Chen, H., Przygienda, T., Cooper, D., Jalil, L., Dontula, S., and G. S. Mishra, "Dynamic Flooding on Dense Graphs", draft-ietf-lsr-dynamic-flooding-08 (work in progress), December 2020.
- [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>>.

Authors' Addresses

Les Ginsberg
Cisco Systems
821 Alder Drive
Milpitas, CA 95035
USA

Email: ginsberg@cisco.com

Peter Psenak
Cisco Systems
Apollo Business Center Mlynske nivy 43
Bratislava 821 09
Slovakia

Email: ppsenak@cisco.com

Marek Karasek
Cisco Systems
Pujmanove 1753/10a, Prague 4 - Nusle
Prague 10 14000
Czech Republic

Email: mkarasek@cisco.com

Acee Lindem
Cisco Systems
301 Midenhall Way
Cary, NC 27513
US

Email: acee@cisco.com

Tony Przygienda
Juniper
1137 Innovation Way
Sunnyvale, Ca
USA

Email: prz@juniper.net

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 24 May 2020

C. Hopps
LabN Consulting, L.L.C.
21 November 2019

YANG Module for IS-IS Reverse Metric
draft-hopps-lsr-yang-isis-reverse-metric-02

Abstract

This document defines a YANG module for managing the reverse metric extension to the the intermediate system to intermediate system routeing protocol.

Status of This Memo

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

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

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

This Internet-Draft will expire on 24 May 2020.

Copyright Notice

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

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

Table of Contents

1. Introduction	2
2. YANG Management	2
2.1. YANG Tree	2
2.2. YANG Module	3
3. IANA Considerations	7
3.1. Updates to the IETF XML Registry	7
3.2. Updates to the YANG Module Names Registry	7
4. Security Considerations	8
5. Normative References	8
6. Informative References	9
Appendix A. Examples	9
A.1. Example Enable XML	9
A.2. Example Use XML	10
A.3. Example JSON	11
Author's Address	12

1. Introduction

This document defines a YANG module for managing the reverse metric extension to the intermediate system to intermediate system routing protocol (IS-IS) [RFC8500], [ISO10589]. Please refer to [RFC8500] for the description and definition of the functionality managed by this module.

The YANG data model described in this document conforms to the Network Management Datastore Architecture defined in [RFC8342].

2. YANG Management

2.1. YANG Tree

The following is the YANG tree diagram ([RFC8340]) for the IS-IS reverse metric extension additions.


```

module: ietf-isis-reverse-metric
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/isis:isis:
      +--rw reverse-metric
        +--rw enable-receive?  boolean
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/isis:isis/isis:interfaces
      /isis:interface:
        +--rw reverse-metric
          +--rw reverse-metric
            +--rw metric?          isis:wide-metric
            +--rw flags
              | +--rw whole-lan?      boolean
              | +--rw allow-unreachable?  boolean
            +--rw exclude-te-metric?  boolean
          +--rw level-1
            +--rw reverse-metric
              +--rw metric?          isis:wide-metric
              +--rw flags
                | +--rw whole-lan?      boolean
                | +--rw allow-unreachable?  boolean
              +--rw exclude-te-metric?  boolean
          +--rw level-2
            +--rw reverse-metric
              +--rw metric?          isis:wide-metric
              +--rw flags
                | +--rw whole-lan?      boolean
                | +--rw allow-unreachable?  boolean
              +--rw exclude-te-metric?  boolean
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/isis:isis/isis:interfaces
      /isis:interface/isis:adjacencies/isis:adjacency:
        +--ro reverse-metric
          +--ro metric?          isis:wide-metric
          +--ro flags
            | +--ro whole-lan?      boolean
            | +--ro allow-unreachable?  boolean
          +--ro te-metric?  uint32

```

2.2. YANG Module

The following is the YANG module for managing the IS-IS reverse metric functionality defined in [RFC8500].


```
<CODE BEGINS> file "ietf-isis-reverse-metric@2019-11-21.yang"
module ietf-isis-reverse-metric {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-isis-reverse-metric";
  prefix isis-rmetric;

  import ietf-routing { prefix "rt"; }
  import ietf-isis { prefix "isis"; }

  organization
    "IETF LSR Working Group (LSR)";

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

    Author: Christian Hopps
            <mailto:chopps@chopps.org>";

  // RFC Ed.: replace XXXX with actual RFC number and
  // remove this note.

  description
    "This module defines the configuration and operational state for
    managing the IS-IS reverse metric functionality [RFC8500].

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

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

    This version of this YANG module is part of RFC XXXX
    (https://tools.ietf.org/html/rfcXXXX); see the RFC itself for
    full legal notices.";

  revision 2019-11-21 {
    description "Initial Revision";
    reference "RFC XXXX: YANG IS-IS Reverse Metric";
  }

  grouping reverse-metric-data {
    description "IS-IS reverse metric data.";
    leaf metric {
```



```
    type isis:wide-metric;
    description "The reverse metric value.";
}

container flags {
    description "The reverse metric flag values.";
    leaf whole-lan {
        type boolean;
        description
            "The 'whole LAN' or W-bit. If true then a DIS processing this
            reverse metric will add the metric value to all the nodes it
            advertises in the pseudo-node LSP for this interface.
            Otherwise it will only increment the metric for the
            advertising node in the pseudo-node LSP for this interface.";
    }
    leaf allow-unreachable {
        type boolean;
        description
            "The 'allow-unreachable' or U-bit. If true it allows the
            neighbor to increment the overall metric up to 2^24-1 rather
            than the lesser maximum of 2^24-2, and if done will cause
            traffic to stop using rather than avoid using the interface.";
    }
}

}

grouping reverse-metric-if-config-data {
    description "IS-IS reverse metric config data.";
    container reverse-metric {
        description "IS-IS reverse metric data.";
        uses reverse-metric-data;
        leaf exclude-te-metric {
            type boolean;
            default false;
            description
                "If true and there is a TE metric defined for this
                interface then do not send the TE metric sub-TLV in the
                reverse metric TLV.";
        }
    }
}

grouping tlv16-reverse-metric {
    description "IS-IS reverse metric TLV data.";
    container reverse-metric {
        description "IS-IS reverse metric TLV data.";
        uses reverse-metric-data;
        leaf te-metric {
```



```
        type uint32;
        description "The TE metric value from the sub-TLV if present.";
    }
}

augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/"
+ "isis:isis" {
    when "../rt:type = 'isis:isis'" {
        description
            "This augment is only valid when routing protocol instance
            type is 'isis'.";
    }

    description
        "The reverse metric configuration for an IS-IS instance.";

    container reverse-metric {
        description "Global reverse metric configuration.";
        leaf enable-receive {
            type boolean;
            default false;
            description
                "Enable handling of reverse metric announcements from
                neighbors. By default reverse metric handling is disabled
                and must be explicitly enabled through this configuration.";
        }
    }
}

augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/"
+ "isis:isis/isis:interfaces/isis:interface" {
    when "../rt:type = 'isis:isis'" {
        description
            "This augment is only valid when routing protocol instance
            type is 'isis'.";
    }

    description
        "The reverse metric configuration for an interface.";

    container reverse-metric {
        description "Announce a reverse metric to neighbors.";
        uses reverse-metric-if-config-data;
        container level-1 {
            description "Announce a reverse metric to level-1 neighbors.";
        }
    }
}
```



```
        uses reverse-metric-if-config-data;
    }
    container level-2 {
        description "Announce a reverse metric to level-2 neighbors.";
        uses reverse-metric-if-config-data;
    }
}
augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/"
+ "isis:isis/isis:interfaces/isis:interface/"
+ "isis:adjacencies/isis:adjacency" {
    when "../.../rt:type = 'isis:isis'" {
        description
            "This augment is only valid when routing protocol instance
            type is 'isis'";
    }

    description
        "The reverse metric state advertised by an adjacency.";
    uses tlv16-reverse-metric;
}
}
<CODE ENDS>
```

3. IANA Considerations

3.1. Updates to the IETF XML Registry

This document registers a URI in the "IETF XML Registry" [RFC3688]. Following the format in [RFC3688], the following registration has been made:

URI	urn:ietf:params:xml:ns:yang:ietf-isis-reverse-metric
-----	--

Registrant Contact	The IESG.
--------------------	-----------

XML	N/A; the requested URI is an XML namespace.
-----	---

3.2. Updates to the YANG Module Names Registry

This document registers one YANG module in the "YANG Module Names" registry [RFC6020]. Following the format in [RFC6020], the following registration has been made:

name	ietf-isis-reverse-metric
------	--------------------------


```
namespace urn:ietf:params:xml:ns:yang:ietf-isis-reverse-metric

prefix isis-rmetric

reference RFC XXXX (RFC Ed.: replace XXX with actual RFC number and
remove this note.)
```

4. Security Considerations

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

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

The YANG module defined in this document can enable, disable and modify the behavior of metrics used by routing. For the security implications regarding these types of changes consult the [RFC8500] which defines the functionality.

5. Normative References

- [ISO10589] International Organization for Standardization, "Intermediate system to intermediate system intra-domain-routing routine information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473)", ISO Standard 10589, 1992.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol

- (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [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>>.
- [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>>.
- [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>>.
- [RFC8500] Shen, N., Amante, S., and M. Abrahamsson, "IS-IS Routing with Reverse Metric", RFC 8500, DOI 10.17487/RFC8500, February 2019, <<https://www.rfc-editor.org/info/rfc8500>>.

6. Informative References

- [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. Examples

A.1. Example Enable XML

Below is an example of YANG XML data to enable reverse metric processing.


```
<rt:routing
  xmlns:rt="urn:ietf:params:xml:ns:yang:ietf-routing"
  xmlns:isis="urn:ietf:params:xml:ns:yang:ietf-isis"
  xmlns:rm="urn:ietf:params:xml:ns:yang:ietf-isis-reverse-metric">
  <rt:control-plane-protocols>
    <rt:control-plane-protocol>
      <rt:type>isis:isis</rt:type>
      <rt:name>default</rt:name>
      <isis:isis>
        <isis:area-address>00</isis:area-address>
        <rm:reverse-metric>
          <rm:enable-receive>true</rm:enable-receive>
        </rm:reverse-metric>
      </isis:isis>
    </rt:control-plane-protocol>
  </rt:control-plane-protocols>
</rt:routing>
```

Figure 1: Example XML data to enable reverse metric processing.

A.2. Example Use XML

Below is an example of YANG XML data for the `ietf-isis-reverse-metric` module.


```
<if:interfaces
  xmlns:if="urn:ietf:params:xml:ns:yang:ietf-interfaces"
  xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type">
  <if:interface>
    <if:name>eth0</if:name>
    <if:type>ianaift:ethernetCsmacd</if:type>
  </if:interface>
</if:interfaces>
<rt:routing
  xmlns:rt="urn:ietf:params:xml:ns:yang:ietf-routing"
  xmlns:isis="urn:ietf:params:xml:ns:yang:ietf-isis"
  xmlns:rm="urn:ietf:params:xml:ns:yang:ietf-isis-reverse-metric">
  <rt:control-plane-protocols>
    <rt:control-plane-protocol>
      <rt:type>isis:isis</rt:type>
      <rt:name>default</rt:name>
      <isis:isis>
        <isis:area-address>00</isis:area-address>
        <isis:interfaces>
          <isis:interface>
            <isis:name>eth0</isis:name>
            <rm:reverse-metric>
              <rm:reverse-metric>
                <rm:metric>
                  65535
                </rm:metric>
              </rm:reverse-metric>
            </rm:reverse-metric>
          </isis:interface>
        </isis:interfaces>
      </isis:isis>
    </rt:control-plane-protocol>
  </rt:control-plane-protocols>
</rt:routing>
```

Figure 2: Example XML data for ietf-isis-reverse-metric module.

A.3. Example JSON

Below is an example of YANG XML data for the ietf-isis-reverse-metric module.


```
{
  "ietf-interfaces:interfaces": {
    "interface": [
      {
        "name": "eth0",
        "type": "iana-if-type:ethernetCsmacd"
      }
    ]
  },
  "ietf-routing:routing": {
    "control-plane-protocols": {
      "control-plane-protocol": [
        {
          "type": "ietf-isis:isis",
          "name": "default",
          "ietf-isis:isis": {
            "area-address": [
              "00"
            ],
            "interfaces": {
              "interface": [
                {
                  "name": "eth0",
                  "ietf-isis-reverse-metric:reverse-metric": {
                    "level-1": {
                      "reverse-metric": {
                        "metric": 65535,
                        "exclude-te-metric": true
                      }
                    }
                  }
                }
              ]
            }
          }
        }
      ]
    }
  }
}
```

Figure 3: Example JSON data for level-1 only reverse metric.

Author's Address

Christian Hopps
LabN Consulting, L.L.C.

Email: chopps@chopps.org

LSR Working Group
Internet-Draft
Intended status: Standards Track
Expires: 13 August 2022

S. Litkowski
Cisco Systems
Y. Qu
Futurewei
P. Sarkar
Individual
I. Chen
The MITRE Corporation
J. Tantsura
Microsoft
9 February 2022

YANG Data Model for IS-IS Segment Routing
draft-ietf-isis-sr-yang-12

Abstract

This document defines a YANG data module that can be used to configure and manage IS-IS Segment Routing, as well as a YANG data module for the management of Signaling Maximum SID Depth (MSD) using IS-IS.

Status of This Memo

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

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

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

This Internet-Draft will expire on 13 August 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document.

Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Overview	2
1.1. Requirements Language	3
1.2. Tree Diagrams	3
2. IS-IS MSD	3
2.1. IS-IS MSD YANG Module	3
3. IS-IS Segment Routing	7
3.1. IS-IS Segment Routing configuration	10
3.1.1. Segment Routing activation	10
3.1.2. Advertising mapping server policy	10
3.1.3. IP Fast reroute	11
3.2. IS-IS Segment Routing YANG Module	11
4. Security Considerations	26
5. Contributors	27
6. Acknowledgements	27
7. IANA Considerations	27
8. Normative References	27
Authors' Addresses	29

1. Overview

YANG [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 module 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.

This document also defines a YANG data module for the management of Signaling Maximum SID Depth (MSD) using IS-IS [RFC8491], which augments the base IS-IS YANG data model.

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

1.1. Requirements Language

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

1.2. Tree Diagrams

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

2. IS-IS MSD

This document defines a module for Signaling Maximum SID Depth (MSD) using IS-IS[RFC8667]. It is an augmentation of the IS-IS base model.

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

```
module: ietf-isis-msd
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/isis:isis/isis:database
      /isis:levels/isis:lsp/isis:router-capabilities:
        +--ro node-msd-tlv
          +--ro node-msds* [msd-type]
            +--ro msd-type      identityref
            +--ro msd-value?    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 link-msd-sub-tlv
            +--ro link-msds* [msd-type]
              +--ro msd-type      identityref
              +--ro msd-value?    uint8
  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 link-msd-sub-tlv
          +--ro link-msds* [msd-type]
            +--ro msd-type      identityref
            +--ro msd-value?    uint8
```

2.1. IS-IS MSD YANG Module


```
<CODE BEGINS> file "ietf-isis-msd@2022-02-09.yang"
module ietf-isis-msd {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-isis-msd";
  prefix isis-msd;

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

  import ietf-isis {
    prefix isis;
  }

  import ietf-mpls-msd {
    prefix mpls-msd;
  }

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

    Author:   Yingzhen Qu
              <mailto:yingzhen.qu@futurewei.com>
    Author:   Acee Lindem
              <mailto:acee@cisco.com>
    Author:   Stephane Litkowski
              <mailto:slitkows.ietf@gmail.com>
    Author:   Jeff Tantsura
              <mailto:jefftant.ietf@gmail.com>

    ";
  description
    "The YANG module augments the base ISIS model to
    manage different types of MSDs.

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

    Copyright (c) 2022 IETF Trust and the persons identified as
    authors of the code.  All rights reserved.

    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject
```


to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>).

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

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.";

```
reference "RFC XXXX: YANG Data Model for OSPF MSD.";

revision 2022-02-09 {
  description
    "Initial Version";
  reference "RFC XXXX: YANG Data Model for ISIS MSD.";
}

grouping link-msd-sub-tlv {
  description
    "Link Maximum SID Depth (MSD) grouping for an interface.";
  container link-msd-sub-tlv {
    list link-msds {
      key "msd-type";
      leaf msd-type {
        type identityref {
          base mpls-msd:msd-base-type;
        }
        description
          "MSD-Types";
      }
      leaf msd-value {
        type uint8;
        description
          "MSD value, in the range of 0-255.";
      }
      description
        "List of link MSDs";
    }
    description
      "Link MSD sub-tlvs.";
  }
}
```



```
/* Node MSD TLV */
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.";
    container node-msd-tlv {
        list node-msds {
            key "msd-type";
            leaf msd-type {
                type identityref {
                    base mpls-msd:msd-base-type;
                }
                description
                    "MSD-Types";
            }
            leaf msd-value {
                type uint8;
                description
                    "MSD value, in the range of 0-255.";
            }
            description
                "Node MSD is the smallest link MSD supported by
                 the node.";
        }
        description
            "Node MSD is the number of SIDs supported by a node.";
        reference
            "RFC 8476: Signaling Maximum SID Depth (MSD) Using OSPF";
    }
}
```

```
/* link MSD sub-tlv */
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 with
      Link MSD sub-TLV.";

    uses link-msd-sub-tlv;
  }

  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 link-msd-sub-tlv;
  }
}
<CODE ENDS>

```

3. 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 prefix-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 prefix-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 prefix-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 prefix-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 prefix-sid-flags
    |   +--ro bits* identityref
    +--ro algorithm?          uint8
    +--ro value                uint32

```

3.1. IS-IS Segment Routing configuration

3.1.1. 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.

3.1.2. 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.

3.1.3. 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.

3.2. IS-IS Segment Routing YANG Module

```
<CODE BEGINS> file "ietf-isis-sr@2022-02-09.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";
    reference
      "RFC 9020 - YANG Data Model for Segment Routing";
  }

  import ietf-segment-routing-mpls {
    prefix "sr-mpls";
    reference
      "RFC 9020 - YANG Data Model for Segment Routing";
  }

  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 8342.

Copyright (c) 2022 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the 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 2022-02-09 {
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 Capability 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 prefix-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-control-plane;
    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>
```

4. Security Considerations

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

The 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 this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

```
/isis:isis/segment-routing
```

```
/isis:isis/protocol-srgb
```

```
/isis:isis/isis:interfaces/isis:interface/segment-routing
```

Some of the readable data nodes in the modules 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.

```
/isis:router-capabilities/sr-capability
```

```
/isis:router-capabilities/sr-algorithms
```



```
/isis:router-capabilities/local-blocks  
  
/isis:router-capabilities/srms-preference  
  
/isis:router-capabilities/node-msd-tlv
```

And the augmentations to the ISIS link state database.

Unauthorized access to any data node of these subtrees can disclose the operational state information of IS-IS protocol on this device.

5. Contributors

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

6. Acknowledgements

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

7. 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: The IESG.  
XML: N/A, the requested URI is an XML namespace
```

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

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

8. Normative References

- [I-D.ietf-isis-yang-isis-cfg]
Litkowski, S., Yeung, D., Lindem, A., Zhang, J., and L. Lhotka, "YANG Data Model for IS-IS Protocol", Work in Progress, Internet-Draft, draft-ietf-isis-yang-isis-cfg-42, 15 October 2019, <<https://www.ietf.org/archive/id/draft-ietf-isis-yang-isis-cfg-42.txt>>.
- [I-D.ietf-spring-sr-yang]
Litkowski, S., Qu, Y., Sarkar, P., and J. Tantsura, "YANG Data Model for Segment Routing", Work in Progress, Internet-Draft, draft-ietf-spring-sr-yang-15, 28 December 2017, <<http://www.ietf.org/internet-drafts/draft-ietf-spring-sr-yang-15.txt>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [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>>.

- [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>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [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>>.
- [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>>.
- [RFC8491] Tantsura, J., Chunduri, U., Aldrin, S., and L. Ginsberg, "Signaling Maximum SID Depth (MSD) Using IS-IS", RFC 8491, DOI 10.17487/RFC8491, November 2018, <<https://www.rfc-editor.org/info/rfc8491>>.
- [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

Yingzhen 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
Microsoft

Email: jefftant.ietf@gmail.com

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: 10 June 2022

T. Li, Ed.
T. Przygienda
Juniper Networks
P. Psenak, Ed.
L. Ginsberg
Cisco Systems, Inc.
H. Chen
Futurewei
D. Cooper
CenturyLink
L. Jalil
Verizon
S. Dontula
ATT
G. Mishra
Verizon Inc.
7 December 2021

Dynamic Flooding on Dense Graphs
draft-ietf-lsr-dynamic-flooding-10

Abstract

Routing with link state protocols in dense network topologies can result in sub-optimal convergence times due to the overhead associated with flooding. This can be addressed by decreasing the flooding topology so that it is less dense.

This document discusses the problem in some depth and an architectural solution. Specific protocol changes for IS-IS, OSPFv2, and OSPFv3 are described in this document.

Status of This Memo

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

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

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

This Internet-Draft will expire on 10 June 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	4
1.1. Requirements Language	5
2. Problem Statement	5
3. Solution Requirements	5
4. Dynamic Flooding	6
4.1. Applicability	7
4.2. Leader election	8
4.3. Computing the Flooding Topology	9
4.4. Topologies on Complete Bipartite Graphs	10
4.4.1. A Minimal Flooding Topology	10
4.4.2. Xia Topologies	10
4.4.3. Optimization	11
4.5. Encoding the Flooding Topology	11
4.6. Advertising the Local Edges Enabled for Flooding	12
5. Protocol Elements	13
5.1. IS-IS TLVs	13
5.1.1. IS-IS Area Leader Sub-TLV	13
5.1.2. IS-IS Dynamic Flooding Sub-TLV	14
5.1.3. IS-IS Area Node IDs TLV	15
5.1.4. IS-IS Flooding Path TLV	16
5.1.5. IS-IS Flooding Request TLV	17
5.1.6. IS-IS LEEF Advertisement	18
5.2. OSPF LSAs and TLVs	18
5.2.1. OSPF Area Leader Sub-TLV	19
5.2.2. OSPF Dynamic Flooding Sub-TLV	20
5.2.3. OSPFv2 Dynamic Flooding Opaque LSA	20
5.2.4. OSPFv3 Dynamic Flooding LSA	22
5.2.5. OSPF Area Router ID TLVs	22
5.2.5.1. OSPFv2 Area Router ID TLV	23
5.2.5.2. OSPFv3 Area Router ID TLV	24

5.2.6.	OSPF Flooding Path TLV	26
5.2.7.	OSPF Flooding Request Bit	27
5.2.8.	OSPF LEEF Advertisement	28
6.	Behavioral Specification	29
6.1.	Terminology	29
6.2.	Flooding Topology	29
6.3.	Leader Election	30
6.4.	Area Leader Responsibilities	30
6.5.	Distributed Flooding Topology Calculation	30
6.6.	Use of LANs in the Flooding Topology	31
6.6.1.	Use of LANs in Centralized mode	31
6.6.2.	Use of LANs in Distributed Mode	31
6.6.2.1.	Partial flooding on a LAN in IS-IS	31
6.6.2.2.	Partial Flooding on a LAN in OSPF	32
6.7.	Flooding Behavior	32
6.8.	Treatment of Topology Events	33
6.8.1.	Temporary Addition of Link to Flooding Topology	33
6.8.2.	Local Link Addition	34
6.8.3.	Node Addition	35
6.8.4.	Failures of Link Not on Flooding Topology	35
6.8.5.	Failures of Link On the Flooding Topology	36
6.8.6.	Node Deletion	36
6.8.7.	Local Link Addition to the Flooding Topology	36
6.8.8.	Local Link Deletion from the Flooding Topology	37
6.8.9.	Treatment of Disconnected Adjacent Nodes	37
6.8.10.	Failure of the Area Leader	37
6.8.11.	Recovery from Multiple Failures	38
6.8.12.	Rate Limiting Temporary Flooding	38
7.	IANA Considerations	39
7.1.	IS-IS	39
7.2.	OSPF	40
7.2.1.	OSPF Dynamic Flooding LSA TLVs Registry	41
7.2.2.	OSPF Link Attributes Sub-TLV Bit Values Registry	42
7.3.	IGP	42
8.	Security Considerations	43
9.	Acknowledgements	43
10.	References	43
10.1.	Normative References	43
10.2.	Informative References	45
	Authors' Addresses	46

1. Introduction

In recent years, there has been increased focus on how to address the dynamic routing of networks that have a bipartite (a.k.a. spine-leaf or leaf-spine), Clos [Clos], or Fat Tree [Leiserson] topology. Conventional Interior Gateway Protocols (IGPs, i.e., IS-IS [ISO10589], OSPFv2 [RFC2328], and OSPFv3 [RFC5340]) under-perform, redundantly flooding information throughout the dense topology, leading to overloaded control plane inputs and thereby creating operational issues. For practical considerations, network architects have resorted to applying unconventional techniques to address the problem, e.g., applying BGP in the data center [RFC7938]. However it is very clear that using an Exterior Gateway Protocol as an IGP is sub-optimal, if only due to the configuration overhead.

The primary issue that is demonstrated when conventional mechanisms are applied is the poor reaction of the network to topology changes. Normal link state routing protocols rely on a flooding algorithm for state distribution within an area. In a dense topology, this flooding algorithm is highly redundant, resulting in unnecessary overhead. Each node in the topology receives each link state update multiple times. Ultimately, all of the redundant copies will be discarded, but only after they have reached the control plane and been processed. This creates issues because significant link state database updates can become queued behind many redundant copies of another update. This delays convergence as the link state database does not stabilize promptly.

In a real world implementation, the packet queues leading to the control plane are necessarily of finite size, so if the flooding rate exceeds the update processing rate for long enough, the control plane will be obligated to drop incoming updates. If these lost updates are of significance, this will further delay stabilization of the link state database and the convergence of the network.

This is not a new problem. Historically, when routing protocols have been deployed in networks where the underlying topology is a complete graph, there have been similar issues. This was more common when the underlying link layer fabric presented the network layer with a full mesh of virtual connections. This was addressed by reducing the flooding topology through IS-IS Mesh Groups [RFC2973], but this approach requires careful configuration of the flooding topology.

Thus, the root problem is not limited to massively scalable data centers. It exists with any dense topology at scale.

This problem is not entirely surprising. Link state routing protocols were conceived when links were very expensive and topologies were sparse. The fact that those same designs are sub-optimal in a dense topology should not come as a huge surprise. The fundamental premise that was addressed by the original designs was an environment of extreme cost and scarcity. Technology has progressed to the point where links are cheap and common. This represents a complete reversal in the economic fundamentals of network engineering. The original designs are to be commended for continuing to provide correct operation to this point, and optimizations for operation in today's environment are to be expected.

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. Problem Statement

In a dense topology, the flooding algorithm that is the heart of conventional link state routing protocols causes a great deal of redundant messaging. This is exacerbated by scale. While the protocol can survive this combination, the redundant messaging is unnecessary overhead and delays convergence. Thus, the problem is to provide routing in dense, scalable topologies with rapid convergence.

3. Solution Requirements

A solution to this problem must then meet the following requirements:

- Requirement 1 Provide a dynamic routing solution. Reachability must be restored after any topology change.
- Requirement 2 Provide a significant improvement in convergence.
- Requirement 3 The solution should address a variety of dense topologies. Just addressing a complete bipartite topology such as K5,8 is insufficient. Multi-stage Clos topologies must also be addressed, as well as topologies that are slight variants. Addressing complete graphs is a good demonstration of generality.
- Requirement 4 There must be no single point of failure. The loss of any link or node should not unduly hinder convergence.

Requirement 5 Dense topologies are subgraphs of much larger topologies. Operational efficiency requires that the dense subgraph not operate in a radically different manner than the remainder of the topology. While some operational differences are permissible, they should be minimized. Changes to nodes outside of the dense subgraph are not acceptable. These situations occur when massively scaled data centers are part of an overall larger wide-area network. Having a second protocol operating just on this subgraph would add much more complexity at the edge of the subgraph where the two protocols would have to inter-operate.

4. Dynamic Flooding

We have observed that the combination of the dense topology and flooding on the physical topology in a scalable network is sub-optimal. However, if we decouple the flooding topology from the physical topology and only flood on a greatly reduced portion of that topology, we can have efficient flooding and retain all of the resilience of existing protocols. A node that supports flooding on the decoupled flooding topology is said to support dynamic flooding.

In this idea, the flooding topology is computed within an IGP area with the dense topology either centrally on an elected node, termed the Area Leader, or in a distributed manner on all nodes that are supporting Dynamic Flooding. If the flooding topology is computed centrally, it is encoded into and distributed as part of the normal link state database. We call this the centralized mode of operation. If the flooding topology is computed in a distributed fashion, we call this the distributed mode of operation. Nodes within such an IGP area would only flood on the flooding topology. On links outside of the normal flooding topology, normal database synchronization mechanisms (i.e., OSPF database exchange, IS-IS CSNPs) would apply, but flooding may not. Details are described in Section 6. New link state information that arrives from outside of the flooding topology suggests that the sender has a different or no flooding topology information and that the link state update should be flooded on the flooding topology as well.

The flooding topology covers the full set of nodes within the area, but excludes some of the links that standard flooding would employ.

Since the flooding topology is computed prior to topology changes, it does not factor into the convergence time and can be done when the topology is stable. The speed of the computation and its distribution, in the case of a centralized mode, is not a significant issue.

If a node does not have any flooding topology information when it receives new link state information, it should flood according to standard flooding rules. This situation will occur when the dense topology is first established, but is unlikely to recur.

When centralized mode is used and if, during a transient, there are multiple flooding topologies being advertised, then nodes should flood link state updates on all of the flooding topologies. Each node should locally evaluate the election of the Area Leader for the IGP area and first flood on its flooding topology. The rationale behind this is straightforward: if there is a transient and there has been a recent change in Area Leader, then propagating topology information promptly along the most likely flooding topology should be the priority.

During transients, it is possible that loops will form in the flooding topology. This is not problematic, as the legacy flooding rules would cause duplicate updates to be ignored. Similarly, during transients, it is possible that the flooding topology may become disconnected. Section 6.8.11 discusses how such conditions are handled.

4.1. Applicability

In a complete graph, this approach is appealing because it drastically decreases the flooding topology without the manual configuration of mesh groups. By controlling the diameter of the flooding topology, as well as the maximum degree node in the flooding topology, convergence time goals can be met and the stability of the control plane can be assured.

Similarly, in a massively scaled data center, where there are many opportunities for redundant flooding, this mechanism ensures that flooding is redundant, with each leaf and spine well connected, while ensuring that no update need make too many hops and that no node shares an undue portion of the flooding effort.

In a network where only a portion of the nodes support Dynamic Flooding, the remaining nodes will continue to perform standard flooding. This is not an issue for correctness, as no node can become isolated.

Flooding that is initiated by nodes that support Dynamic Flooding will remain within the flooding topology until it reaches a legacy node, which will resume legacy flooding. Standard flooding will be bounded by nodes supporting Dynamic Flooding, which can help limit the propagation of unnecessary flooding. Whether or not the network can remain stable in this condition is unknown and may be very dependent on the number and location of the nodes that support Dynamic Flooding.

During incremental deployment of dynamic flooding an area will consist of one or more sets of connected nodes that support dynamic flooding and one or more sets of connected nodes that do not, i.e., nodes that support standard flooding. The flooding topology is the union of these sets of nodes. Each set of nodes that does not support dynamic flooding needs to be part of the flooding topology and such a set of nodes may provide connectivity between two or more sets of nodes that support dynamic flooding.

4.2. Leader election

A single node within the dense topology is elected as an Area Leader.

A generalization of the mechanisms used in existing Designated Router (OSPF) or Designated Intermediate-System (IS-IS) elections suffices. The elected node is known as the Area Leader.

In the case of centralized mode, the Area Leader is responsible for computing and distributing the flooding topology. When a new Area Leader is elected and has distributed new flooding topology information, then any prior Area Leaders should withdraw any of their flooding topology information from their link state database entries.

In the case of distributed mode, the distributed algorithm advertised by the Area Leader **MUST** be used by all nodes that participate in Dynamic Flooding.

Not every node needs to be a candidate to be Area Leader within an area, as a single candidate is sufficient for correct operation. For redundancy, however, it is strongly **RECOMMENDED** that there be multiple candidates.

4.3. Computing the Flooding Topology

There is a great deal of flexibility in how the flooding topology may be computed. For resilience, it needs to at least contain a cycle of all nodes in the dense subgraph. However, additional links could be added to decrease the convergence time. The trade-off between the density of the flooding topology and the convergence time is a matter for further study. The exact algorithm for computing the flooding topology in the case of the centralized computation need not be standardized, as it is not an interoperability issue. Only the encoding of the result needs to be documented. In the case of distributed mode, all nodes in the IGP area need to use the same algorithm to compute the flooding topology. It is possible to use private algorithms to compute flooding topology, so long as all nodes in the IGP area use the same algorithm.

While the flooding topology should be a covering cycle, it need not be a Hamiltonian cycle where each node appears only once. In fact, in many relevant topologies this will not be possible e.g., K5,8. This is fortunate, as computing a Hamiltonian cycle is known to be NP-complete.

A simple algorithm to compute the topology for a complete bipartite graph is to simply select unvisited nodes on each side of the graph until both sides are completely visited. If the number of nodes on each side of the graph are unequal, then revisiting nodes on the less populated side of the graph will be inevitable. This algorithm can run in $O(N)$ time, so is quite efficient.

While a simple cycle is adequate for correctness and resiliency, it may not be optimal for convergence. At scale, a cycle may have a diameter that is half the number of nodes in the graph. This could cause an undue delay in link state update propagation. Therefore it may be useful to have a bound on the diameter of the flooding topology. Introducing more links into the flooding topology would reduce the diameter, but at the trade-off of possibly adding redundant messaging. The optimal trade-off between convergence time and graph diameter is for further study.

Similarly, if additional redundancy is added to the flooding topology, specific nodes in that topology may end up with a very high degree. This could result in overloading the control plane of those nodes, resulting in poor convergence. Thus, it may be optimal to have an upper bound on the degree of nodes in the flooding topology. Again, the optimal trade-off between graph diameter, node degree, and convergence time, and topology computation time is for further study.

If the leader chooses to include a multi-node broadcast LAN segment as part of the flooding topology, all of the connectivity to that LAN segment should be included as well. Once updates are flooded onto the LAN, they will be received by every attached node.

4.4. Topologies on Complete Bipartite Graphs

Complete bipartite graph topologies have become popular for data center applications and are commonly called leaf-spine or spine-leaf topologies. In this section, we discuss some flooding topologies that are of particular interest in these networks.

4.4.1. A Minimal Flooding Topology

We define a Minimal Flooding Topology on a complete bipartite graph as one in which the topology is connected and each node has at least degree two. This is of interest because it guarantees that the flooding topology has no single points of failure.

In practice, this implies that every leaf node in the flooding topology will have a degree of two. As there are usually more leaves than spines, the degree of the spines will be higher, but the load on the individual spines can be evenly distributed.

This type of flooding topology is also of interest because it scales well. As the number of leaves increases, we can construct flooding topologies that perform well. Specifically, for n spines and m leaves, if $m \geq n(n/2 - 1)$, then there is a flooding topology that has a diameter of four.

4.4.2. Xia Topologies

We define a Xia Topology on a complete bipartite graph as one in which all spine nodes are bi-connected through leaves with degree two, but the remaining leaves all have degree one and are evenly distributed across the spines.

Constructively, we can create a Xia topology by iterating through the spines. Each spine can be connected to the next spine by selecting any unused leaf. Since leaves are connected to all spines, all leaves will have a connection to both the first and second spine and we can therefore choose any leaf without loss of generality. Continuing this iteration across all of the spines, selecting a new leaf at each iteration, will result in a path that connects all spines. Adding one more leaf between the last and first spine will produce a cycle of n spines and n leaves.

At this point, $m-n$ leaves remain unconnected. These can be distributed evenly across the remaining spines, connected by a single link.

Xia topologies represent a compromise that trades off increased risk and decreased performance for lower flooding amplification. Xia topologies will have a larger diameter. For m spines, the diameter will be $m + 2$.

In a Xia topology, some leaves are singly connected. This represents a risk in that in some failures, convergence may be delayed. However, there may be some alternate behaviors that can be employed to mitigate these risks. If a leaf node sees that its single link on the flooding topology has failed, it can compensate by performing a database synchronization check with a different spine. Similarly, if a leaf determines that its connected spine on the flooding topology has failed, it can compensate by performing a database synchronization check with a different spine. In both of these cases, the synchronization check is intended to ameliorate any delays in link state propagation due to the fragmentation of the flooding topology.

The benefit of this topology is that flooding load is easily understood. Each node in the spine cycle will never receive an update more than twice. For m leaves and n spines, a spine never transmits more than $(m/n + 1)$ updates.

4.4.3. Optimization

If two nodes are adjacent on the flooding topology and there are a set of parallel links between them, then any given update MUST be flooded over a single one of those links. Selection of the specific link is implementation specific.

4.5. Encoding the Flooding Topology

There are a variety of ways that the flooding topology could be encoded efficiently. If the topology was only a cycle, a simple list of the nodes in the topology would suffice. However, this is insufficiently flexible as it would require a slightly different encoding scheme as soon as a single additional link is added. Instead, we choose to encode the flooding topology as a set of intersecting paths, where each path is a set of connected edges.

Advertisement of the flooding topology includes support for multi-access LANs. When a LAN is included in the flooding topology, all edges between the LAN and nodes connected to the LAN are assumed to be part of the flooding topology. In order to reduce the size of the

flooding topology advertisement, explicit advertisement of these edges is optional. Note that this may result in the possibility of "hidden nodes" existing which are actually part of the flooding topology but which are not explicitly mentioned in the flooding topology advertisements. These hidden nodes can be found by examination of the Link State database where connectivity between a LAN and nodes connected to the LAN is fully specified.

Note that while all nodes **MUST** be part of the advertised flooding topology not all multi-access LANs need to be included. Only those LANs which are part of the flooding topology need to be included in the advertised flooding topology.

Other encodings are certainly possible. We have attempted to make a useful trade off between simplicity, generality, and space.

4.6. Advertising the Local Edges Enabled for Flooding

Correct operation of the flooding topology requires that all nodes which participate in the flooding topology choose local links for flooding which are consistent with the calculated flooding topology. Failure to do so could result in unexpected partition of the flooding topology and/or sub-optimal flooding reduction. As an aid to diagnosing problems when dynamic flooding is in use, this document defines a means of advertising what local edges are enabled for flooding (LEEF). The protocol specific encodings are defined in Sections 5.1.6 and 5.2.8.

The following guidelines apply:

Advertisement of LEEFs is optional.

As the flooding topology is defined by edges (not by links), in cases where parallel adjacencies to the same neighbor exist, the advertisement **SHOULD** indicate that all such links have been enabled.

LEEF advertisements **MUST NOT** include edges enabled for temporary flooding (Section 6.7).

LEEF advertisements **MUST NOT** be used either when calculating a flooding topology or when determining what links to add temporarily to the flooding topology when the flooding topology is temporarily partitioned.

5. Protocol Elements

5.1. IS-IS TLVs

The following TLVs/sub-TLVs are added to IS-IS:

1. A sub-TLV that an IS may inject into its LSP to indicate its preference for becoming Area Leader.
2. A sub-TLV that an IS may inject into its LSP to indicate that it supports Dynamic Flooding and the algorithms that it supports for distributed mode, if any.
3. A TLV to carry the list of system IDs that compromise the flooding topology for the area.
4. A TLV to carry a path which is part of the flooding topology
5. A TLV that requests flooding from the adjacent node

5.1.1. IS-IS Area Leader Sub-TLV

The Area Leader Sub-TLV allows a system to:

1. Indicate its eligibility and priority for becoming Area Leader.
2. Indicate whether centralized or distributed mode is to be used to compute the flooding topology in the area.
3. Indicate the algorithm identifier for the algorithm that is used to compute the flooding topology in distributed mode.

Intermediate Systems (nodes) that are not advertising this Sub-TLV are not eligible to become Area Leader.

The Area Leader is the node with the numerically highest Area Leader priority in the area. In the event of ties, the node with the numerically highest system ID is the Area Leader. Due to transients during database flooding, different nodes may not agree on the Area Leader.

The Area Leader Sub-TLV is advertised as a Sub-TLV of the IS-IS Router Capability TLV-242 that is defined in [RFC7981] and 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								Priority								Algorithm							

Type: TBD1

Length: 2

Priority: 0-255, unsigned integer

Algorithm: a numeric identifier in the range 0-255 that identifies the algorithm used to calculate the flooding topology. The following values are defined:

- 0: Centralized computation by the Area Leader.
- 1-127: Standardized distributed algorithms. Individual values are to be assigned according to the "Specification Required" policy defined in [RFC8126] (see Section 7.3).
- 128-254: Private distributed algorithms. Individual values are to be assigned according to the "Private Use" policy defined in [RFC8126] (see Section 7.3).
- 255: Reserved

5.1.2. IS-IS Dynamic Flooding Sub-TLV

The Dynamic Flooding Sub-TLV allows a system to:

1. Indicate that it supports Dynamic Flooding. This is indicated by the advertisement of this Sub-TLV.
2. Indicate the set of algorithms that it supports for distributed mode, if any.

In incremental deployments, understanding which nodes support Dynamic Flooding can be used to optimize the flooding topology. In distributed mode, knowing the capabilities of the nodes can allow the Area Leader to select the optimal algorithm.

The Dynamic Flooding Sub-TLV is advertised as a Sub-TLV of the IS-IS Router Capability TLV (242) [RFC7981] and 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... |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Type: TBD7

Length: 0-255; number of Algorithms

Algorithm: zero or more numeric identifiers in the range 0-255 that identifies the algorithm used to calculate the flooding topology, as described in Section 5.1.1.

5.1.3. IS-IS Area Node IDs TLV

The IS-IS Area Node IDs TLV is only used in centralized mode.

The Area Node IDs TLV is used by the Area Leader to enumerate the Node IDs (System ID + pseudo-node ID) that it has used in computing the area flooding topology. Conceptually, the Area Leader creates a list of node IDs for all nodes in the area (including pseudo-nodes for all LANs in the topology), assigning indices to each node, starting with index 0.

Because the space in a single TLV is limited, more than one TLV may be required to encode all of the node IDs in the area. This TLV may be present in multiple LSPs.

The format of the Area Node IDs TLV is:

```

      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      | Starting Index |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|L| Reserved     | Node IDs ...
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
Node IDs continued ....
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Type: TBD2

Length: 3 + ((System ID Length + 1) * (number of node IDs))

Starting index: The index of the first node ID that appears in this TLV.

L (Last): This bit is set if the index of the last node ID that appears in this TLV is equal to the last index in the full list of node IDs for the area.

Node IDs: A concatenated list of node IDs for the area

If there are multiple IS-IS Area Node IDs TLVs with the L bit set advertised by the same node, the TLV which specifies the smaller maximum index is used and the other TLV(s) with L bit set are ignored. TLVs which specify node IDs with indices greater than that specified by the TLV with the L bit set are also ignored.

5.1.1.4. IS-IS Flooding Path TLV

IS-IS Flooding Path TLV is only used in centralized mode.

The Flooding Path TLV is used to denote a path in the flooding topology. The goal is an efficient encoding of the links of the topology. A single link is a simple case of a path that only covers two nodes. A connected path may be described as a sequence of indices: (I1, I2, I3, ...), denoting a link from the system with index 1 to the system with index 2, a link from the system with index 2 to the system with index 3, and so on.

If a path exceeds the size that can be stored in a single TLV, then the path may be distributed across multiple TLVs by the replication of a single system index.

Complex topologies that are not a single path can be described using multiple TLVs.

The Flooding Path TLV contains a list of system indices relative to the systems advertised through the Area Node IDs TLV. At least 2 indices must be included in the TLV. Due to the length restriction of TLVs, this TLV can contain at most 126 system indices.

The Flooding Path TLV has the 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									Starting Index																	
Index 2									Additional indices ...																										

Type: TBD3

Length: 2 * (number of indices in the path)

Starting index: The index of the first system in the path.

Index 2: The index of the next system in the path.

Additional indices (optional): A sequence of additional indices to systems along the path.

5.1.5. IS-IS Flooding Request TLV

The Flooding Request TLV allows a system to request an adjacent node to enable flooding towards it on a specific link in the case where the connection to adjacent node is not part of the existing flooding topology.

Nodes that support Dynamic Flooding MAY include the Flooding Request TLV in its IIH PDUs.

The Flooding Request TLV has the format:

0									1									2									3											
0	1	2	3	4	5	6	7	8	0	1	2	3	4	5	6	7	8	0	1	2	3	4	5	6	7	8	0	1	2	3	4	5	6	7	8	9	0	1
Type									Length									Levels									R Scope											
R ...																																						

Type: TBD9

Length: 1 + number of advertised Flooding Scopes

Levels - the level(s) for which flooding is requested. Levels are encoded as the circuit type specified in IS-IS [ISO10589]

R bit: MUST be 0 and is ignored on receipt.

Scope: Flooding Scope for which the flooding is requested as defined by LSP Flooding Scope Identifier Registry defined by [RFC7356]. Inclusion of flooding scopes is optional and is only necessary if [RFC7356] is supported. Multiple flooding scopes MAY be included.

Circuit Flooding Scope MUST NOT be sent in the Flooding Request TLV and MUST be ignored if received.

When the TLV is received in a level specific LAN-Hello PDU (L1-LAN-IIH or L2-LAN-IIH) only levels which match the PDU type are valid. Levels which do not match the PDU type MUST be ignored on receipt.

When the TLV is received in a Point-to-Point Hello (P2P-IIH) only levels which are supported by the established adjacency are valid. Levels which are not supported by the adjacency MUST be ignored on receipt.

If flooding was disabled on the received link due to Dynamic Flooding, then flooding MUST be temporarily enabled over the link for the specified Circuit Type(s) and Flooding Scope(s) received in the Flooding Request TLV. Flooding MUST be enabled until the Circuit Type or Flooding Scope is no longer advertised in the Flooding Request TLV or the TLV no longer appears in IIH PDUs received on the link.

When the flooding is temporarily enabled on the link for any Circuit Type or Flooding Scope due to received Flooding Request TLV, the receiver MUST perform standard database synchronization for the corresponding Circuit Type(s) and Flooding Scope(s) on the link. In the case of IS-IS, this results in setting SRM bit for all related LSPs on the link and sending CSNPs.

So long as the Flooding Request TLV is being received flooding MUST NOT be disabled for any of the Circuit Types or Flooding Scopes present in the Flooding Request TLV even if the connection between the neighbors is removed from the flooding topology. Flooding for such Circuit Types or Flooding Scopes MUST continue on the link and be considered as temporarily enabled.

5.1.6. IS-IS LEEF Advertisement

In support of advertising which edges are currently enabled in the flooding topology, an implementation MAY indicate that a link is part of the flooding topology by advertising a bit value in the Link Attributes sub-TLV defined by [RFC5029].

The following bit value is defined by this document:

Local Edge Enabled for Flooding (LEEF) - suggested value 4 (to be assigned by IANA)

5.2. OSPF LSAs and TLVs

This section defines new LSAs and TLVs for both OSPFv2 and OSPFv3.

Following objects are added:

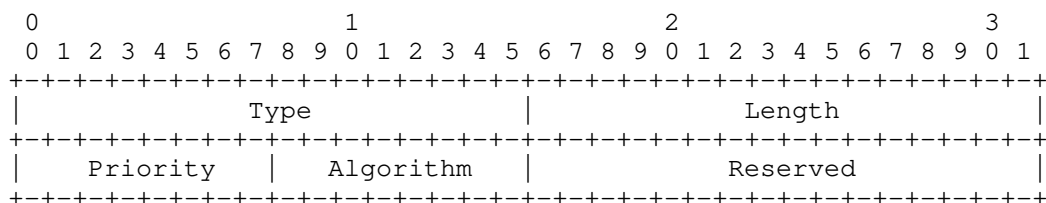
1. A TLV that is used to advertise the preference for becoming Area Leader.
2. A TLV that is used to indicate the support for Dynamic Flooding and the algorithms that the advertising node supports for distributed mode, if any.
3. OSPFv2 Opaque LSA and OSPFv3 LSA to advertise the flooding topology for centralized mode.
4. A TLV to carry the list of Router IDs that comprise the flooding topology for the area.
5. A TLV to carry a path which is part of the flooding topology.
6. The bit in the LLS Type 1 Extended Options and Flags requests flooding from the adjacent node.

5.2.1. OSPF Area Leader Sub-TLV

The usage of the OSPF Area Leader Sub-TLV is identical to IS-IS and is described in Section 5.1.1.

The OSPF Area Leader Sub-TLV is used by both OSPFv2 and OSPFv3.

The OSPF Area Leader Sub-TLV is advertised as a top-level TLV of the RI LSA that is defined in [RFC7770] and has the following format:



Type: TBD4

Length: 4 octets

Priority: 0-255, unsigned integer

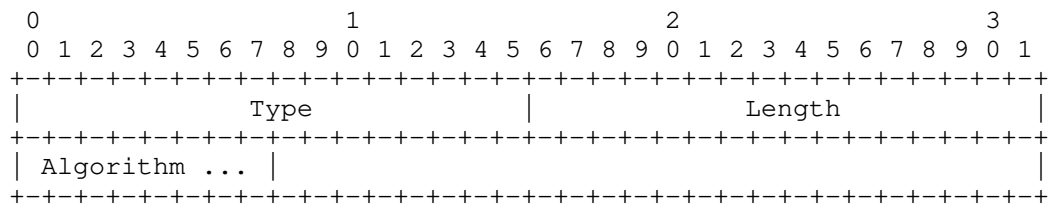
Algorithm: as defined in Section 5.1.1.

5.2.2. OSPF Dynamic Flooding Sub-TLV

The usage of the OSPF Dynamic Flooding Sub-TLV is identical to IS-IS and is described in Section 5.1.2.

The OSPF Dynamic Flooding Sub-TLV is used by both OSPFv2 and OSPFv3.

The OSPF Dynamic Flooding Sub-TLV is advertised as a top-level TLV of the RI LSA that is defined in [RFC7770] and has the following format:



Type: TBD8

Length: number of Algorithms

Algorithm: as defined in Section 5.1.1.

5.2.3. OSPFv2 Dynamic Flooding Opaque LSA

The OSPFv2 Dynamic Flooding Opaque LSA is only used in centralized mode.

The OSPFv2 Dynamic Flooding Opaque LSA is used to advertise additional data related to the dynamic flooding in OSPFv2. OSPFv2 Opaque LSAs are described in [RFC5250].

Multiple OSPFv2 Dynamic Flooding Opaque LSAs can be advertised by an OSPFv2 router. The flooding scope of the OSPFv2 Dynamic Flooding Opaque LSA is area-local.

The format of the OSPFv2 Dynamic Flooding Opaque LSA is as follows:

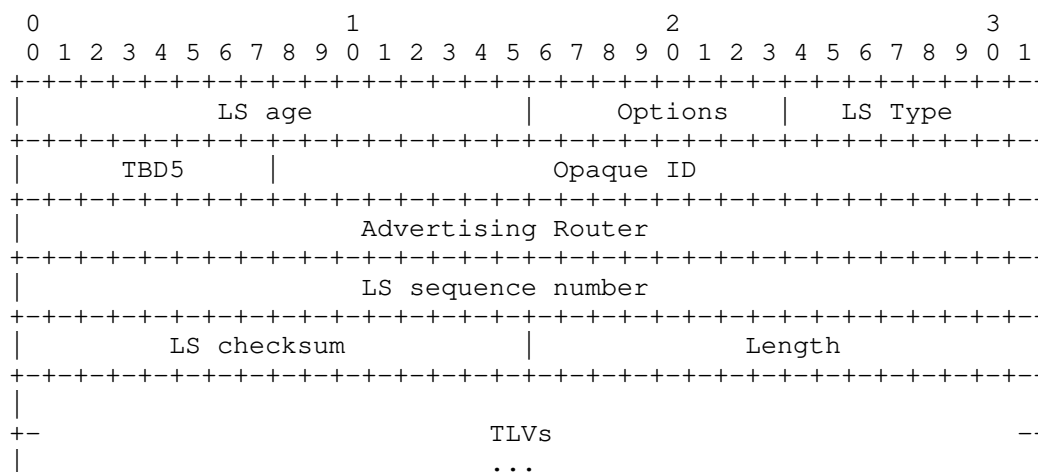


Figure 1: OSPFv2 Dynamic Flooding Opaque LSA

The opaque type used by OSPFv2 Dynamic Flooding Opaque LSA is TBD. The opaque type is used to differentiate the various type of OSPFv2 Opaque LSAs and is described in section 3 of [RFC5250]. The LS Type is 10. The LSA Length field [RFC2328] represents the total length (in octets) of the Opaque LSA including the LSA header and all TLVs (including padding).

The Opaque ID field is an arbitrary value used to maintain multiple Dynamic Flooding Opaque LSAs. For OSPFv2 Dynamic Flooding Opaque LSAs, the Opaque ID has no semantic significance other than to differentiate Dynamic Flooding Opaque LSAs originated by the same OSPFv2 router.

The format of the TLVs within the body of the OSPFv2 Dynamic Flooding Opaque LSA is the same as the format used by the Traffic Engineering Extensions to OSPF [RFC3630].

The Length field defines the length of the value portion in octets (thus a TLV with no value portion would have a length of 0). The TLV is padded to 4-octet alignment; padding is not included in the length field (so a 3-octet value would have a length of 3, but the total size of the TLV would be 8 octets). Nested TLVs are also 32-bit aligned. For example, a 1-octet value would have the length field set to 1, and 3 octets of padding would be added to the end of the value portion of the TLV. The padding is composed of zeros.

5.2.4. OSPFv3 Dynamic Flooding LSA

The OSPFv3 Dynamic Flooding Opaque LSA is only used in centralized mode.

The OSPFv3 Dynamic Flooding LSA is used to advertise additional data related to the dynamic flooding in OSPFv3.

The OSPFv3 Dynamic Flooding LSA has a function code of TBD. The flooding scope of the OSPFv3 Dynamic Flooding LSA is area-local. The U bit will be set indicating that the OSPFv3 Dynamic Flooding LSA should be flooded even if it is not understood. The Link State ID (LSID) value for this LSA is the Instance ID. OSPFv3 routers MAY advertise multiple Dynamic Flooding Opaque LSAs in each area.

The format of the OSPFv3 Dynamic Flooding LSA is as follows:

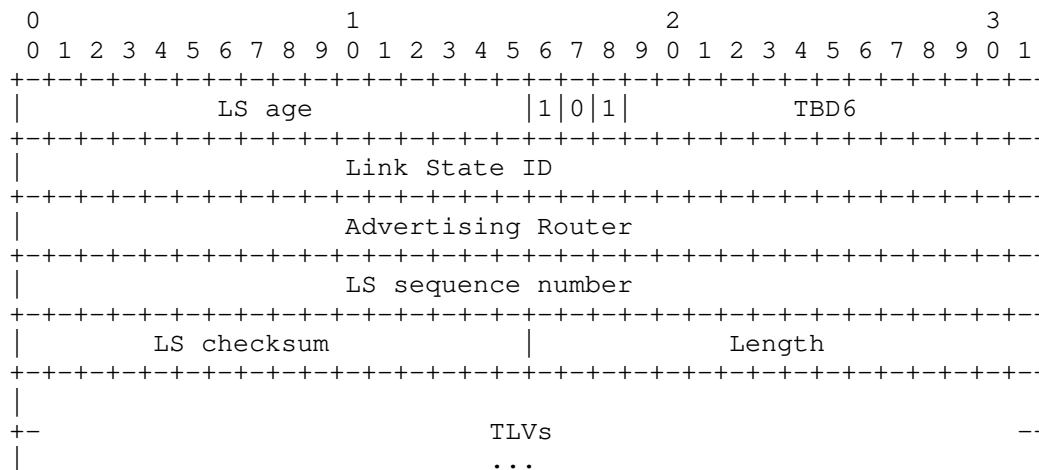


Figure 2: OSPFv3 Dynamic Flooding LSA

5.2.5. OSPF Area Router ID TLVs

In OSPF new TLVs are introduced to advertise indeces associated with nodes and Broadcast/NBMA networks. Due to identifier differences between OSPFv2 and OSPFv3 two different TLVs are defined as decribed in the following sub-sections.

The OSPF Area Router ID TLVs are used by the Area Leader to enumerate the Router IDs that it has used in computing the flooding topology. This includes the identifiers associated with Broadcast/NBMA networks as defined for Network LSAs. Conceptually, the Area Leader creates a list of Router IDs for all routers in the area, assigning indices to each router, starting with index 0.

5.2.5.1. OSPFv2 Area Router ID TLV

This TLV is a top level TLV of the OSPFv2 Dynamic Flooding Opaque LSA.

Because the space in a single OSPFv2 Area Router IDs TLV is limited, more than one TLV may be required to encode all of the Router IDs in the area. This TLV may also occur in multiple OSPFv2 Dynamic Flooding Opaque LSAs so that all Router IDs can be advertised.

Each entry in the OSPFv2 Area Router IDs TLV represents either a node or a Broadcast/NBMA network identifier. An entry has the following format:

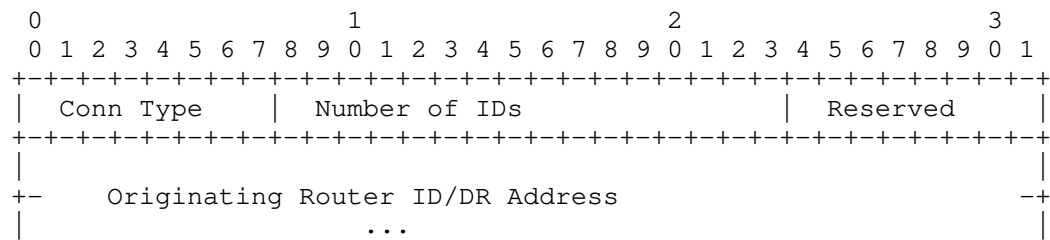


Figure 3: OSPFv2 Router IDs TLV Entry

Conn Type: 1 byte

- The following values are defined:

- 1 - Router
- 2 - Designated Router

Number of IDs: 2 bytes

Reserved: 1 byte, MUST be transmitted as 0 and MUST be ignored on receipt

Originating Router ID/DR Address: (4 * Number of IDs) bytes as indicated by the ID Type

The format of the Area Router IDs TLV is:

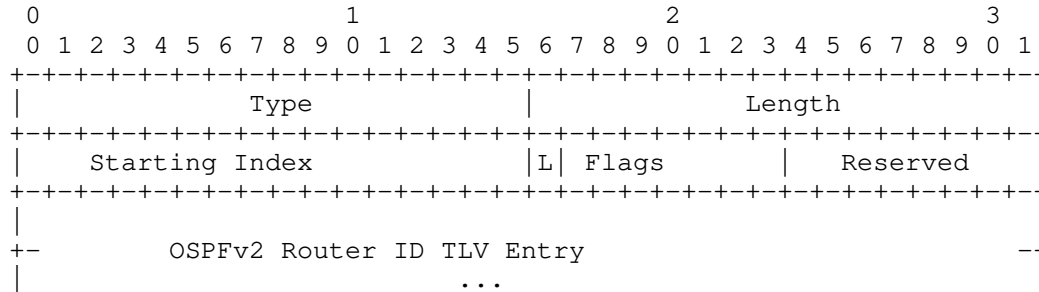


Figure 4: OSPFv2 Area Router IDs TLV

TLV Type: 1

TLV Length: 4 + (8 * the number TLV entries)

Starting index: The index of the first Router/Designated Router ID that appears in this TLV.

L (Last): This bit is set if the index of the last Router/Designated ID that appears in this TLV is equal to the last index in the full list of Router IDs for the area.

OSPFv2 Router ID TLV Entries: A concatenated list of Router ID TLV Entries for the area.

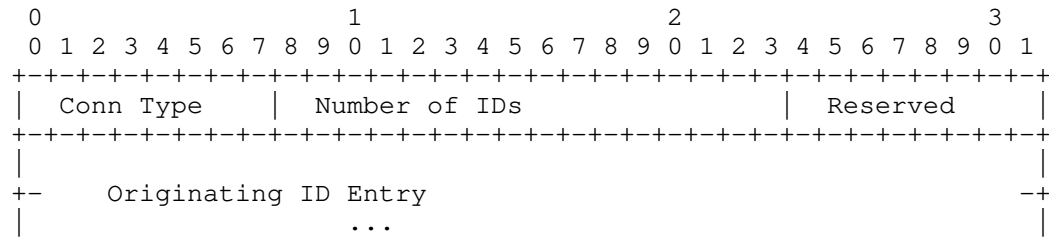
If there are multiple OSPFv2 Area Router ID TLVs with the L bit set advertised by the same router, the TLV which specifies the smaller maximum index is used and the other TLV(s) with L bit set are ignored. TLVs which specify Router IDs with indices greater than that specified by the TLV with the L bit set are also ignored.

5.2.5.2. OSPFv3 Area Router ID TLV

This TLV is a top level TLV of the OSPFv3 Dynamic Flooding LSA.

Because the space in a single OSPFv3 Area Router ID TLV is limited, more than one TLV may be required to encode all of the Router IDs in the area. This TLV may also occur in multiple OSPFv3 Dynamic Flooding Opaque LSAs so that all Router IDs can be advertised.

Each entry in the OSPFv3 Area Router IDs TLV represents either a router or a Broadcast/NBMA network identifier. An entry has the following format:



where

Conn Type - 1 byte

The following values are defined:

1 - Router

2 - Designated Router

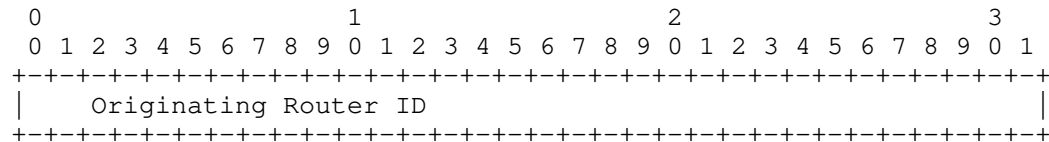
Number of IDs - 2 bytes

Reserved - 1 byte

MUST be transmitted as 0 and MUST be ignored on receipt

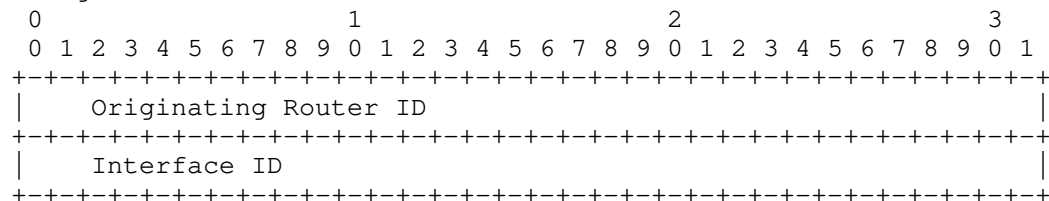
Originating ID Entry takes one of the following forms:

Router:



Length of Originating ID Entry is 4 * Number of IDs) bytes

Designated Router:



Length of Originating ID Entry is (8 * Number of IDs) bytes

Figure 5: OSPFv3 Router ID TLV Entry

The format of the OSPFv3Area Router IDs TLV is:

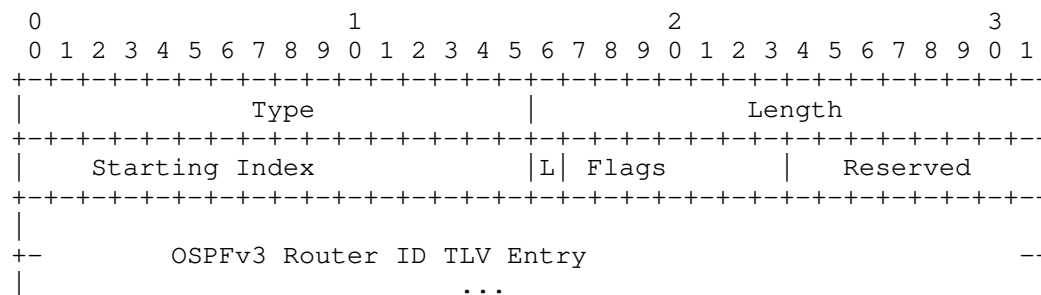


Figure 6: OSPFv3 Area Router IDs TLV

TLV Type: 1

TLV Length: 4 + sum of the lengths of all TLV entries

Starting index: The index of the first Router/Designated Router ID that appears in this TLV.

L (Last): This bit is set if the index of the last Router/Designated Router ID that appears in this TLV is equal to the last index in the full list of Router IDs for the area.

OSPFv3 Router ID TLV Entries: A concatenated list of Router ID TLV Entries for the area.

If there are multiple OSPFv3 Area Router ID TLVs with the L bit set advertised by the same router, the TLV which specifies the smaller maximum index is used and the other TLV(s) with L bit set are ignored. TLVs which specify Router IDs with indices greater than that specified by the TLV with the L bit set are also ignored.

5.2.6. OSPF Flooding Path TLV

The OSPF Flooding Path TLV is a top level TLV of the OSPFv2 Dynamic Flooding Opaque LSAs and OSPFv3 Dynamic Flooding LSA.

The usage of the OSPF Flooding Path TLV is identical to IS-IS and is described in Section 5.1.4.

The OSPF Flooding Path TLV contains a list of Router ID indices relative to the Router IDs advertised through the OSPF Area Router IDs TLV. At least 2 indices must be included in the TLV.

Multiple OSPF Flooding Path TLVs can be advertised in a single OSPFv2 Dynamic Flooding Opaque LSA or OSPFv3 Dynamic Flooding LSA. OSPF Flooding Path TLVs can also be advertised in multiple OSPFv2 Dynamic Flooding Opaque LSAs or OSPFv3 Dynamic Flooding LSA, if they all can not fit in a single LSA.

The Flooding Path TLV has the format:

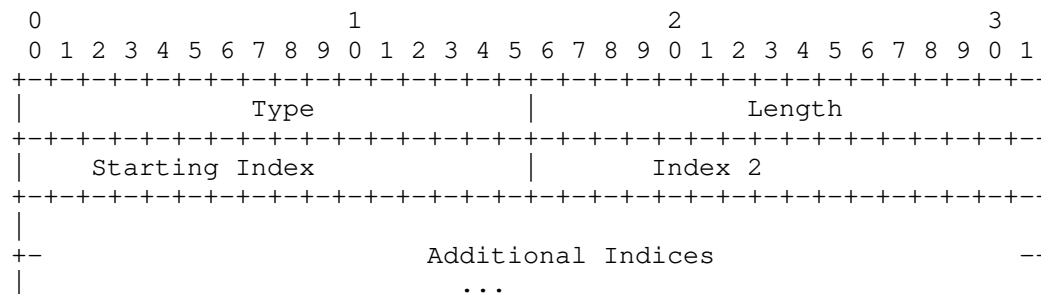


Figure 7: OSPF Flooding Path TLV

TLV Type: 2

TLV Length: 2 * (number of indices in the path)

Starting index: The index of the first Router ID in the path.

Index 2: The index of the next Router ID in the path.

Additional indices (optional): A sequence of additional indices to Router IDs along the path.

5.2.7. OSPF Flooding Request Bit

A single new option bit, the Flooding-Request (FR-bit), is defined in the LLS Type 1 Extended Options and Flags field [RFC2328]. The FR-bit allows a router to request an adjacent node to enable flooding towards it on a specific link in the case where the connection to adjacent node is not part of the current flooding topology.

Nodes that support Dynamic Flooding MAY include FR-bit in its OSPF LLS Extended Options and Flags TLV.

If FR-bit is signalled for an area for which the flooding on the link was disabled due to Dynamic Flooding, the flooding MUST be temporarily enabled over such link and area. Flooding MUST be enabled until FR-bit is no longer advertised in the OSPF LLS Extended Options and Flags TLV or the OSPF LLS Extended Options and Flags TLV no longer appears in the OSPF Hellos.

When the flooding is temporarily enabled on the link for any area due to received FR-bit in OSPF LLS Extended Options and Flags TLV, the receiver MUST perform standard database synchronization for the corresponding area(s) on the link. If the adjacency is already in the FULL state, mechanism specified in [RFC4811] MUST be used for database resynchronization.

So long as the FR-bit is being received in the OSPF LLS Extended Options and Flags TLV for an area, flooding MUST NOT be disabled in such area even if the connection between the neighbors is removed from the flooding topology. Flooding for such area MUST continue on the link and be considered as temporarily enabled.

5.2.8. OSPF LEEF Advertisement

In support of advertising which edges are currently enabled in the flooding topology, an implementation MAY indicate that a link is part of the flooding topology. The OSPF Link Attributes Bits TLV is defined to support this advertisement.

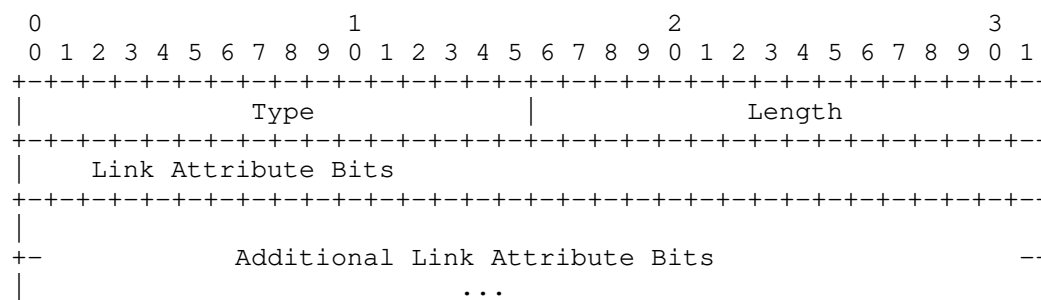


Figure 8: OSPF Link Attributes Bits TLV

Type: TBD and specific to OSPFv2 and OSPFv3

Length: size of the Link Attribute Bits in bytes. It MUST be a multiple of 4 bytes.

The following bits are defined:

Bit #0: - Local Edge Enabled for Flooding (LEEF)

OSPF Link-attribute Bits TLV appears as:

1. a sub-TLV of the OSPFv2 Extended Link TLV [RFC7684]
2. a sub-TLV of the OSPFv3 Router-Link TLV [RFC8362]

6. Behavioral Specification

In this section, we specify the detailed behaviors of the nodes participating in the IGP.

6.1. Terminology

We define some terminology here that is used in the following sections:

A node is considered reachable if it is part of the connected network graph. Note that this is independent of any constraints which may be considered when performing IGP SPT calculation (e.g., link metrics, OL bit state, etc.). Two-way-connectivity check **MUST** be performed before including an edge in the connected network graph.

Node is connected to the flooding topology, if it has at least one local link, which is part of the flooding topology.

Node is disconnected from the flooding topology when it is not connected to the flooding topology.

Current flooding topology - latest version of the flooding topology received (in case of the centralized mode) or calculated locally (in case of the distributed mode).

6.2. Flooding Topology

The flooding topology **MUST** include all reachable nodes in the area.

If a node's reachability changes, the flooding topology **MUST** be recalculated. In centralized mode, the Area Leader **MUST** advertise a new flooding topology.

If a node becomes disconnected from the current flooding topology but is still reachable then a new flooding topology **MUST** be calculated. In centralized mode the Area Leader **MUST** advertise the new flooding topology.

The flooding topology **SHOULD** be bi-connected.

6.3. Leader Election

Any node that is capable MAY advertise its eligibility to become Area Leader.

Nodes that are not reachable are not eligible as Area Leader. Nodes that do not advertise their eligibility to become Area Leader are not eligible. Amongst the eligible nodes, the node with the numerically highest priority is the Area Leader. If multiple nodes all have the highest priority, then the node with the numerically highest system identifier in the case of IS-IS, or Router-ID in the case of OSPFv2 and OSPFv3 is the Area Leader.

6.4. Area Leader Responsibilities

If the Area Leader operates in centralized mode, it MUST advertise algorithm 0 in its Area Leader Sub-TLV. In order for Dynamic Flooding to be enabled it also MUST compute and advertise a flooding topology for the area. The Area Leader may update the flooding topology at any time, however, it should not destabilize the network with undue or overly frequent topology changes. If the Area Leader operates in centralized mode and needs to advertise a new flooding topology, it floods the new flooding topology on both the new and old flooding topologies.

If the Area Leader operates in distributed mode, it MUST advertise a non-zero algorithm in its Area Leader Sub-TLV.

When the Area Leader advertises algorithm 0 in its Area Leader Sub-TLV and does not advertise a flooding topology, Dynamic Flooding is disabled for the area. Note this applies whether the Area Leader intends to operate in centralized mode or in distributed mode.

Note that once Dynamic Flooding is enabled, disabling it risks destabilizing the network.

6.5. Distributed Flooding Topology Calculation

If the Area Leader advertises a non-zero algorithm in its Area Leader Sub-TLV, all nodes in the area that support Dynamic Flooding and the value of algorithm advertised by the Area Leader MUST compute the flooding topology based on the Area Leader's advertised algorithm.

Nodes that do not support the value of algorithm advertised by the Area Leader MUST continue to use standard flooding mechanism as defined by the protocol.

Nodes that do not support the value of algorithm advertised by the Area Leader MUST be considered as Dynamic Flooding incapable nodes by the Area Leader.

If the value of the algorithm advertised by the Area Leader is from the range 128-254 (private distributed algorithms), it is the responsibility of the network operator to guarantee that all nodes in the area have a common understanding of what the given algorithm value represents.

6.6. Use of LANs in the Flooding Topology

Use of LANs in the flooding topology differs depending on whether the area is operating in Centralized or Distributed mode.

6.6.1. Use of LANs in Centralized mode

As specified in Section 4.5, when a LAN is advertised as part of the flooding topology, all nodes connected to the LAN are assumed to be using the LAN as part of the flooding topology. This assumption is made to reduce the size of the Flooding Topology advertisement.

6.6.2. Use of LANs in Distributed Mode

In distributed mode, the flooding topology is NOT advertised, therefore the space consumed to advertise it is not a concern. It is therefore possible to assign only a subset of the nodes connected to the LAN to use the LAN as part of the flooding topology. Doing so may further optimize flooding by reducing the amount of redundant flooding on a LAN. However, support of flooding only by a subset of the nodes connected to a LAN requires some modest - but backwards compatible - changes in the way flooding is performed on a LAN.

6.6.2.1. Partial flooding on a LAN in IS-IS

Designated Intermediate System (DIS) for a LAN MUST use standard flooding behavior.

Non-DIS nodes whose connection to the LAN is included in the flooding topology MUST use standard flooding behavior.

Non-DIS nodes whose connection to the LAN is NOT included in the flooding topology behave as follows:

- * Received CSNPs from the DIS are ignored
- * PSNPs are NOT originated on the LAN

- * LSAs received on the LAN which are newer than the corresponding LSP present in the LSPDB are retained and flooded on all local circuits which are part of the flooding topology (i.e., do not discard newer LSAs simply because they were received on a LAN which the receiving node is not using for flooding)
- * LSAs received on the LAN which are older or same as the corresponding LSP present in the LSPDB are silently discarded
- * LSAs received on links other than the LAN are NOT flooded on the LAN

NOTE: If any node connected to the LAN requests the enablement of temporary flooding all nodes revert to standard flooding behavior.

6.6.2.2. Partial Flooding on a LAN in OSPF

Designated Router (DR) and Backup Designated Router (BDR) for LANs MUST use standard flooding behavior.

Non-DR/BDR nodes whose connection to the LAN is included in the flooding topology use standard flooding behavior.

Non-DR/BDR nodes whose connection to the LAN is NOT included in the flooding topology behave as follows:

- * LSAs received on the LAN are acknowledged to DR/BDR
- * LSAs received on interfaces other than the LAN are NOT flooded on the LAN

NOTE: If any node connected to the LAN requests the enablement of temporary flooding all nodes revert to standard flooding behavior.

NOTE: The sending of LSA acks by nodes NOT using the LAN as part of the flooding topology eliminates the need for changes on the part of the DR/BDR - which might Include nodes which do not support the flooding optimizations.

6.7. Flooding Behavior

Nodes that support Dynamic Flooding MUST use the flooding topology for flooding when possible, and MUST NOT revert to standard flooding when a valid flooding topology is available.

In some cases a node that supports Dynamic Flooding may need to add a local link(s) to the flooding topology temporarily, even though the link(s) is not part of the calculated flooding topology. This is termed "temporary flooding" and is discussed in Section 6.8.1.

The flooding topology is calculated locally in the case of distributed mode. In centralized mode the flooding topology is advertised in the area link state database. Received link state updates, whether received on a link that is in the flooding topology or on a link that is not in the flooding topology, **MUST** be flooded on all links that are in the flooding topology, except for the link on which the update was received.

In centralized mode, if multiple flooding topologies are present in the area link state database, the node **SHOULD** flood on each of these topologies.

When the flooding topology changes on a node, either as a result of the local computation in distributed mode or as a result of the advertisement from the Area Leader in centralized mode, the node **MUST** continue to flood on both the old and new flooding topology for a limited amount of time. This is required to provide all nodes sufficient time to migrate to the new flooding topology.

6.8. Treatment of Topology Events

In this section, we explicitly consider a variety of different topological events in the network and how Dynamic Flooding should address them.

6.8.1. Temporary Addition of Link to Flooding Topology

In some cases a node that supports Dynamic Flooding may need to add a local link(s) to the flooding topology temporarily, even though the link(s) is not part of the calculated flooding topology. We refer to this as "temporary flooding" on the link.

When temporary flooding is enabled on the link, the flooding needs to be enabled from both directions on the link. To achieve that, the following steps **MUST** be performed:

Link State Database needs to be re-synchronised on the link. This is done using the standard protocol mechanisms. In the case of IS-IS, this results in setting SRM bit for all LSPs on the circuit and sending complete set of CSNPs on it. In OSPF, the mechanism specified in [RFC4811] is used.

Flooding is enabled locally on the link.

Flooding is requested from the neighbor using the mechanism specified in section Section 5.1.5 or Section 5.2.7.

The request for temporary flooding is withdrawn on the link when all of the following conditions are met:

- Node itself is connected to the current flooding topology.

- Adjacent node is connected to the current flooding topology.

Any change in the flooding topology MUST result in evaluation of the above conditions for any link on which the temporary flooding was enabled.

Temporary flooding is stopped on the link when both adjacent nodes stop requesting temporary flooding on the link.

6.8.2. Local Link Addition

If a local link is added to the topology, the protocol will form a normal adjacency on the link and update the appropriate link state advertisements for the nodes on either end of the link. These link state updates will be flooded on the flooding topology.

In centralized mode, the Area Leader, upon receiving these updates, may choose to retain the existing flooding topology or may choose to modify the flooding topology. If it elects to change the flooding topology, it will update the flooding topology in the link state database and flood it using the new flooding topology.

In distributed mode, any change in the topology, including the link addition, MUST trigger the flooding topology recalculation. This is done to ensure that all nodes converge to the same flooding topology, regardless of the time of the calculation.

Temporary flooding MUST be enabled on the newly added local link, if at least one of the following conditions are met:

- The node on which the local link was added is not connected to the current flooding topology.

- The new adjacent node is not connected to the current flooding topology.

Note that in this case there is no need to perform a database synchronization as part of the enablement of the temporary flooding, because it has been part of the adjacency bring-up itself.

If multiple local links are added to the topology before the flooding topology is updated, temporary flooding MUST be enabled on a subset of these links.

6.8.3. Node Addition

If a node is added to the topology, then at least one link is also added to the topology. Section 6.8.2 applies.

A node which has a large number of neighbors is at risk for introducing a local flooding storm if all neighbors are brought up at once and temporary flooding is enabled on all links simultaneously. The most robust way to address this is to limit the rate of initial adjacency formation following bootup. This both reduces unnecessary redundant flooding as part of initial database synchronization and minimizes the need for temporary flooding as it allows time for the new node to be added to the flooding topology after only a small number of adjacencies have been formed.

In the event a node elects to bring up a large number of adjacencies simultaneously, a significant amount of redundant flooding may be introduced as multiple neighbors of the new node enable temporary flooding to the new node which initially is not part of the flooding topology.

6.8.4. Failures of Link Not on Flooding Topology

If a link that is not part of the flooding topology fails, then the adjacent nodes will update their link state advertisements and flood them on the flooding topology.

In centralized mode, the Area Leader, upon receiving these updates, may choose to retain the existing flooding topology or may choose to modify the flooding topology. If it elects to change the flooding topology, it will update the flooding topology in the link state database and flood it using the new flooding topology.

In distributed mode, any change in the topology, including the failure of the link that is not part of the flooding topology MUST trigger the flooding topology recalculation. This is done to ensure that all nodes converge to the same flooding topology, regardless of the time of the calculation.

6.8.5. Failures of Link On the Flooding Topology

If there is a failure on the flooding topology, the adjacent nodes will update their link state advertisements and flood them. If the original flooding topology is bi-connected, the flooding topology should still be connected despite a single failure.

If the failed local link represented the only connection to the flooding topology on the node where the link failed, the node **MUST** enable temporary flooding on a subset of its local links. This allows the node to send its updated link state advertisement(s) and also keep receiving link state updates from other nodes in the network before the new flooding topology is calculated and distributed (in the case of centralized mode).

In centralized mode, the Area Leader will notice the change in the flooding topology, recompute the flooding topology, and flood it using the new flooding topology.

In distributed mode, all nodes supporting dynamic flooding will notice the change in the topology and recompute the new flooding topology.

6.8.6. Node Deletion

If a node is deleted from the topology, then at least one link is also removed from the topology. Section 6.8.4 and Section 6.8.5 apply.

6.8.7. Local Link Addition to the Flooding Topology

If the new flooding topology is received in the case of centralized mode, or calculated locally in the case of distributed mode and the local link on the node that was not part of the flooding topology has been added to the flooding topology, the node **MUST**:

Re-synchronize the Link State Database over the link. This is done using the standard protocol mechanisms. In the case of IS-IS, this results in setting SRM bit for all LSPs on the circuit and sending a complete set of CSNPs. In OSPF, the mechanism specified in [RFC4811] is used.

Make the link part of the flooding topology and start flooding over it

6.8.8. Local Link Deletion from the Flooding Topology

If the new flooding topology is received in the case of centralized mode, or calculated locally in the case of distributed mode and the local link on the node that was part of the flooding topology has been removed from the flooding topology, the node MUST remove the link from the flooding topology.

The node MUST keep flooding on such link for a limited amount of time to allow other nodes to migrate to the new flooding topology.

If the removed local link represented the only connection to the flooding topology on the node, the node MUST enable temporary flooding on a subset of its local links. This allows the node to send its updated link state advertisement(s) and also keep receiving link state updates from other nodes in the network before the new flooding topology is calculated and distributed (in the case of centralized mode).

6.8.9. Treatment of Disconnected Adjacent Nodes

Every time there is a change in the flooding topology a node MUST check if there are any adjacent nodes that are disconnected from the current flooding topology. Temporary flooding MUST be enabled towards a subset of the disconnected nodes.

6.8.10. Failure of the Area Leader

The failure of the Area Leader can be detected by observing that it is no longer reachable. In this case, the Area Leader election process is repeated and a new Area Leader is elected.

In order to minimize disruption to Dynamic Flooding if the Area Leader becomes unreachable, the node which has the second highest priority for becoming Area Leader (including the system identifier/Router-ID tie breaker if necessary) SHOULD advertise the same algorithm in its Area Leader Sub-TLV as the Area Leader and (in centralized mode) SHOULD advertise a flooding topology. This SHOULD be done even when the Area Leader is reachable.

In centralized mode, the new Area Leader will compute a new flooding topology and flood it using the new flooding topology. To minimize disruption, the new flooding topology SHOULD have as much in common as possible with the old flooding topology. This will minimize the risk of over-flooding.

In the distributed mode, the new flooding topology will be calculated on all nodes that support the algorithm that is advertised by the new Area Leader. Nodes that do not support the algorithm advertised by the new Area Leader will no longer participate in Dynamic Flooding and will revert to standard flooding.

6.8.11. Recovery from Multiple Failures

In the unlikely event of multiple failures on the flooding topology, it may become partitioned. The nodes that remain active on the edges of the flooding topology partitions will recognize this and will try to repair the flooding topology locally by enabling temporary flooding towards the nodes that they consider disconnected from the flooding topology until a new flooding topology becomes connected again.

Nodes where local failure was detected update their own link state advertisements and flood them on the remainder of the flooding topology.

In centralized mode, the Area Leader will notice the change in the flooding topology, recompute the flooding topology, and flood it using the new flooding topology.

In distributed mode, all nodes that actively participate in Dynamic Flooding will compute the new flooding topology.

Note that this is very different from the area partition because there is still a connected network graph between the nodes in the area. The area may remain connected and forwarding may still be effective.

6.8.12. Rate Limiting Temporary Flooding

As discussed in the previous sections, there are events which require the introduction of temporary flooding on edges which are not part of the current flooding topology. This can occur regardless of whether the area is operating in centralized mode or distributed mode.

Nodes which decide to enable temporary flooding also have to decide whether to do so on a subset of the edges which are currently not part of the flooding topology or on all the edges which are currently not part of the flooding topology. Doing the former risks a longer convergence time as it is possible that the initial set of edges enabled does not fully repair the flooding topology. Doing the latter risks introducing a flooding storm which destabilizes the network.

It is recommended that a node implement rate limiting on the number of edges on which it chooses to enable temporary flooding. Initial values for the number of edges to enable and the rate at which additional edges may subsequently be enabled is left as an implementation decision.

7. IANA Considerations

7.1. IS-IS

This document requests the following code points from the "sub-TLVs for TLV 242" registry (IS-IS Router CAPABILITY TLV).

Type: TBD1

Description: IS-IS Area Leader Sub-TLV

Reference: This document (Section 5.1.1)

Type: TBD7

Description: IS-IS Dynamic Flooding Sub-TLV

Reference: This document (Section 5.1.2)

This document requests that IANA allocate and assign code points from the "IS-IS TLV Codepoints" registry. One for each of the following TLVs:

Type: TBD2

Description: IS-IS Area System IDs TLV

Reference: This document (Section 5.1.3)

Type: TBD3

Description: IS-IS Flooding Path TLV

Reference: This document (Section 5.1.4)

Type: TBD9

Description: IS-IS Flooding Request TLV

Reference: This document (Section 5.1.5)

This document requests that IANA allocate a new bit value from the "link-attribute bit values for sub-TLV 19 of TLV 22" registry.

Local Edge Enabled for Flooding (LEEF) - suggested value 4 (to be assigned by IANA)

7.2. OSPF

This document requests the following code points from the "OSPF Router Information (RI) TLVs" registry:

Type: TBD4

Description: OSPF Area Leader Sub-TLV

Reference: This document (Section 5.2.1)

Type: TBD8

Description: OSPF Dynamic Flooding Sub-TLV

Reference: This document (Section 5.2.2)

This document requests the following code point from the "Opaque Link-State Advertisements (LSA) Option Types" registry:

Type: TBD5

Description: OSPFv2 Dynamic Flooding Opaque LSA

Reference: This document (Section 5.2.3)

This document requests the following code point from the "OSPFv3 LSA Function Codes" registry:

Type: TBD6

Description: OSPFv3 Dynamic Flooding LSA

Reference: This document (Section 5.2.4)

This document requests a new bit in LLS Type 1 Extended Options and Flags registry:

Bit Position: TBD10

Description: Flooding Request bit

Reference: This document (Section 5.2.7)

This document requests the following code point from the "OSPFv2 Extended Link TLV Sub-TLVs" registry:

Type: TBD11

Description: OSPFv2 Link Attributes Bits Sub-TLV

Reference: This document (Section 5.2.8)

This document requests the following code point from the "OSPFv3 Extended LSA Sub-TLVs" registry:

Type: TBD12

Description: OSPFv3 Link Attributes Bits Sub-TLV

Reference: This document (Section 5.2.8)

7.2.1. OSPF Dynamic Flooding LSA TLVs Registry

This specification also requests a new registry - "OSPF Dynamic Flooding LSA TLVs". New values can be allocated via IETF Review or IESG Approval

The "OSPF Dynamic Flooding LSA TLVs" registry will define top-level TLVs for the OSPFv2 Dynamic Flooding Opaque LSA and OSPFv3 Dynamic Flooding LSAs. It should be added to the "Open Shortest Path First (OSPF) Parameters" registries group.

The following initial values are allocated:

Type: 0

Description: Reserved

Reference: This document

Type: 1

Description: OSPF Area Router IDs TLV

Reference: This document (Section 5.2.5)

Type: 2

Description: OSPF Flooding Path TLV

Reference: This document (Section 5.2.6)

Types in the range 32768-33023 are for experimental use; these will not be registered with IANA, and MUST NOT be mentioned by RFCs.

Types in the range 33024-65535 are not to be assigned at this time. Before any assignments can be made in the 33024-65535 range, there MUST be an IETF specification that specifies IANA Considerations that covers the range being assigned.

7.2.2. OSPF Link Attributes Sub-TLV Bit Values Registry

This specification also requests a new registry - "OSPF Link Attributes Sub-TLV Bit Values". New values can be allocated via IETF Review or IESG Approval

The "OSPF Link Attributes Sub-TLV Bit Values" registry defines Link Attribute bit values for the OSPFv2 Link Attributes Sub-TLV and OSPFv3 Link Attributes Sub-TLV. It should be added to the "Open Shortest Path First (OSPF) Parameters" registries group.

The following initial value is allocated:

Bit Number: 0

Description: Local Edge Enabled for Flooding(LEEF)

Reference: This document (Section 5.2.8)

7.3. IGP

IANA is requested to set up a registry called "IGP Algorithm Type For Computing Flooding Topology" under an existing "Interior Gateway Protocol (IGP) Parameters" IANA registries.

Values in this registry come from the range 0-255.

The initial values in the IGP Algorithm Type For Computing Flooding Topology registry are:

0: Reserved for centralized mode.

1-127: Available for standards action. Individual values are to be assigned according to the "Specification Required" policy defined in [RFC8126].

128-254: Reserved for private use.

255: Reserved.

8. Security Considerations

This document introduces no new security issues. Security of routing within a domain is already addressed as part of the routing protocols themselves. This document proposes no changes to those security architectures.

It is possible that an attacker could become Area Leader and introduce a flawed flooding algorithm into the network thus compromising the operation of the protocol. Authentication methods as describe in [RFC5304] and [RFC5310] for IS-IS, [RFC2328] and [RFC7474] for OSPFv2 and [RFC5340] and [RFC4552] for OSPFv3 SHOULD be used to prevent such attack.

9. Acknowledgements

The authors would like to thank Sarah Chen for her contribution to this work.

The authors would like to thank Zeqing (Fred) Xia, Naiming Shen, Adam Sweeney and Olufemi Komolafe for their helpful comments.

The authors would like to thank Tom Edsall for initially introducing them to the problem.

Advertising Local Edges Enabled for Flooding (LEEF) is based on an idea proposed in [I-D.cc-lsr-flooding-reduction]. We wish to thank the authors of that draft.

10. References

10.1. Normative References

- [ISO10589] International Organization for Standardization, "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)", ISO/IEC 10589:2002, October 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>>.

- [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <<https://www.rfc-editor.org/info/rfc2328>>.
- [RFC4552] Gupta, M. and N. Melam, "Authentication/Confidentiality for OSPFv3", RFC 4552, DOI 10.17487/RFC4552, June 2006, <<https://www.rfc-editor.org/info/rfc4552>>.
- [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>>.
- [RFC5250] Berger, L., Bryskin, I., Zinin, A., and R. Coltun, "The OSPF Opaque LSA Option", RFC 5250, DOI 10.17487/RFC5250, July 2008, <<https://www.rfc-editor.org/info/rfc5250>>.
- [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>>.
- [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <<https://www.rfc-editor.org/info/rfc5340>>.
- [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>>.
- [RFC7474] Bhatia, M., Hartman, S., Zhang, D., and A. Lindem, Ed., "Security Extension for OSPFv2 When Using Manual Key Management", RFC 7474, DOI 10.17487/RFC7474, April 2015, <<https://www.rfc-editor.org/info/rfc7474>>.
- [RFC7684] Psenak, P., Gredler, H., Shakir, R., Henderickx, W., Tantsura, J., and A. Lindem, "OSPFv2 Prefix/Link Attribute Advertisement", RFC 7684, DOI 10.17487/RFC7684, November 2015, <<https://www.rfc-editor.org/info/rfc7684>>.
- [RFC7770] Lindem, A., Ed., Shen, N., Vasseur, JP., Aggarwal, R., and S. Shaffer, "Extensions to OSPF for Advertising Optional Router Capabilities", RFC 7770, DOI 10.17487/RFC7770, February 2016, <<https://www.rfc-editor.org/info/rfc7770>>.

- [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>>.
- [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>>.
- [RFC8362] Lindem, A., Roy, A., Goethals, D., Reddy Vallem, V., and F. Baker, "OSPFv3 Link State Advertisement (LSA) Extensibility", RFC 8362, DOI 10.17487/RFC8362, April 2018, <<https://www.rfc-editor.org/info/rfc8362>>.

10.2. Informative References

- [Clos] Clos, C., "A Study of Non-Blocking Switching Networks", The Bell System Technical Journal Vol. 32(2), DOI 10.1002/j.1538-7305.1953.tb01433.x, March 1953, <<http://dx.doi.org/10.1002/j.1538-7305.1953.tb01433.x>>.
- [I-D.cc-lsr-flooding-reduction]
Chen, H., Toy, M., Yang, Y., Wang, A., Liu, X., Fan, Y., and L. Liu, "Flooding Topology Computation Algorithm", Work in Progress, Internet-Draft, draft-cc-lsr-flooding-reduction-09, 5 June 2020, <<https://www.ietf.org/archive/id/draft-cc-lsr-flooding-reduction-09.txt>>.
- [Leiserson]
Leiserson, C. E., "Fat-Trees: Universal Networks for Hardware-Efficient Supercomputing", IEEE Transactions on Computers 34(10):892-901, 1985.
- [RFC2973] Balay, R., Katz, D., and J. Parker, "IS-IS Mesh Groups", RFC 2973, DOI 10.17487/RFC2973, October 2000, <<https://www.rfc-editor.org/info/rfc2973>>.
- [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, DOI 10.17487/RFC3630, September 2003, <<https://www.rfc-editor.org/info/rfc3630>>.
- [RFC4811] Nguyen, L., Roy, A., and A. Zinin, "OSPF Out-of-Band Link State Database (LSDB) Resynchronization", RFC 4811, DOI 10.17487/RFC4811, March 2007, <<https://www.rfc-editor.org/info/rfc4811>>.

[RFC7938] Lapukhov, P., Premji, A., and J. Mitchell, Ed., "Use of BGP for Routing in Large-Scale Data Centers", RFC 7938, DOI 10.17487/RFC7938, August 2016, <<https://www.rfc-editor.org/info/rfc7938>>.

Authors' Addresses

Tony Li (editor)
Juniper Networks
1133 Innovation Way
Sunnyvale, California 94089
United States of America

Email: tony.li@tony.li

Tony Przygienda
Juniper Networks
1133 Innovation Way
Sunnyvale, California 94089
United States of America

Email: prz@juniper.net

Peter Psenak (editor)
Cisco Systems, Inc.
Eurovea Centre, Central 3
Pribinova Street 10
81109 Bratislava
Slovakia

Email: ppsenak@cisco.com

Les Ginsberg
Cisco Systems, Inc.
510 McCarthy Blvd.
Milpitas, California 95035
United States of America

Email: ginsberg@cisco.com

Huaimo Chen
Futurewei
Boston, Ma,
United States of America

Email: hchen@futurewei.com

Dave Cooper
CenturyLink
1025 Eldorado Blvd
Broomfield, Colorado 80021
United States of America

Email: Dave.Cooper@centurylink.com

Luay Jalil
Verizon
Richardson, Texas 75081
United States of America

Email: luay.jalil@verizon.com

Srinath Dontula
ATT
200 S Laurel Ave
Middletown, New Jersey 07748
United States of America

Email: sd947e@att.com

Gyan S. Mishra
Verizon Inc.
13101 Columbia Pike
Silver Spring, Maryland 20904
United States of America

Phone: 301 502-1347
Email: gyan.s.mishra@verizon.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 9, 2022

P. Psenak, Ed.
Cisco Systems
S. Hegde
Juniper Networks, Inc.
C. Filsfils
Cisco Systems, Inc.
K. Talaulikar
Arrcus, Inc
A. Gulko
Edward Jones
April 7, 2022

IGP Flexible Algorithm
draft-ietf-lsr-flex-algo-19

Abstract

IGP protocols traditionally compute best paths over the network based on the IGP metric assigned to the links. Many network deployments use RSVP-TE based or Segment Routing based Traffic Engineering to steer traffic over a path that is computed using different metrics or constraints than the shortest IGP path. This document proposes a solution that allows IGPs themselves to compute constraint-based paths over the network. This document also specifies a way of using Segment Routing (SR) Prefix-SIDs and SRv6 locators to steer packets along the constraint-based paths.

Status of This Memo

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

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

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

This Internet-Draft will expire on October 9, 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
2. Requirements Language	4
3. Terminology	4
4. Flexible Algorithm	5
5. Flexible Algorithm Definition Advertisement	6
5.1. IS-IS Flexible Algorithm Definition Sub-TLV	6
5.2. OSPF Flexible Algorithm Definition TLV	8
5.3. Common Handling of Flexible Algorithm Definition TLV	9
6. Sub-TLVs of IS-IS FAD Sub-TLV	10
6.1. IS-IS Flexible Algorithm Exclude Admin Group Sub-TLV	11
6.2. IS-IS Flexible Algorithm Include-Any Admin Group Sub-TLV	12
6.3. IS-IS Flexible Algorithm Include-All Admin Group Sub-TLV	12
6.4. IS-IS Flexible Algorithm Definition Flags Sub-TLV	13
6.5. IS-IS Flexible Algorithm Exclude SRLG Sub-TLV	14
7. Sub-TLVs of OSPF FAD TLV	15
7.1. OSPF Flexible Algorithm Exclude Admin Group Sub-TLV	15
7.2. OSPF Flexible Algorithm Include-Any Admin Group Sub-TLV	16
7.3. OSPF Flexible Algorithm Include-All Admin Group Sub-TLV	16
7.4. OSPF Flexible Algorithm Definition Flags Sub-TLV	16
7.5. OSPF Flexible Algorithm Exclude SRLG Sub-TLV	17
8. IS-IS Flexible Algorithm Prefix Metric Sub-TLV	18
9. OSPF Flexible Algorithm Prefix Metric Sub-TLV	19
10. OSPF Flexible Algorithm ASBR Reachability Advertisement	21
10.1. OSPFv2 Extended Inter-Area ASBR LSA	21
10.1.1. OSPFv2 Extended Inter-Area ASBR TLV	23
10.2. OSPF Flexible Algorithm ASBR Metric Sub-TLV	23
11. Advertisement of Node Participation in a Flex-Algorithm	25
11.1. Advertisement of Node Participation for Segment Routing	26
11.2. Advertisement of Node Participation for Other Applications	26
12. Advertisement of Link Attributes for Flex-Algorithm	26

13. Calculation of Flexible Algorithm Paths	27
13.1. Multi-area and Multi-domain Considerations	29
14. Flex-Algorithm and Forwarding Plane	31
14.1. Segment Routing MPLS Forwarding for Flex-Algorithm	32
14.2. SRv6 Forwarding for Flex-Algorithm	32
14.3. Other Applications' Forwarding for Flex-Algorithm	33
15. Operational Considerations	33
15.1. Inter-area Considerations	33
15.2. Usage of SRLG Exclude Rule with Flex-Algorithm	34
15.3. Max-metric consideration	35
16. Backward Compatibility	35
17. Security Considerations	35
18. IANA Considerations	36
18.1. IGP IANA Considerations	36
18.1.1. IGP Algorithm Types Registry	36
18.1.2. IGP Metric-Type Registry	36
18.2. Flexible Algorithm Definition Flags Registry	37
18.3. IS-IS IANA Considerations	37
18.3.1. Sub TLVs for Type 242	37
18.3.2. Sub TLVs for for TLVs 135, 235, 236, and 237	37
18.3.3. Sub-Sub-TLVs for Flexible Algorithm Definition Sub-TLV	37
18.4. OSPF IANA Considerations	38
18.4.1. OSPF Router Information (RI) TLVs Registry	38
18.4.2. OSPFv2 Extended Prefix TLV Sub-TLVs	39
18.4.3. OSPFv3 Extended-LSA Sub-TLVs	39
18.4.4. OSPF Flex-Algorithm Prefix Metric Bits	39
18.4.5. OSPF Opaque LSA Option Types	39
18.4.6. OSPFv2 Extended Inter-Area ASBR TLVs	40
18.4.7. OSPFv2 Inter-Area ASBR Sub-TLVs	40
18.4.8. OSPF Flexible Algorithm Definition TLV Sub-TLV Registry	40
18.4.9. Link Attribute Applications Registry	42
19. Acknowledgements	42
20. References	42
20.1. Normative References	42
20.2. Informative References	44
Authors' Addresses	46

1. Introduction

An IGP-computed path based on the shortest IGP metric is often be replaced by a traffic-engineered path due to the traffic requirements which are not reflected by the IGP metric. Some networks engineer the IGP metric assignments in a way that the IGP metric reflects the link bandwidth or delay. If, for example, the IGP metric is reflecting the bandwidth on the link and the application traffic is

delay sensitive, the best IGP path may not reflect the best path from such an application's perspective.

To overcome this limitation, various sorts of traffic engineering have been deployed, including RSVP-TE and SR-TE, in which case the TE component is responsible for computing paths based on additional metrics and/or constraints. Such paths need to be installed in the forwarding tables in addition to, or as a replacement for, the original paths computed by IGPs. Tunnels are often used to represent the engineered paths and mechanisms like one described in [RFC3906] are used to replace the native IGP paths with such tunnel paths.

This document specifies a set of extensions to IS-IS, OSPFv2, and OSPFv3 that enable a router to advertise TLVs that (a) identify calculation-type, (b) specify a metric-type, and (c) describe a set of constraints on the topology, that are to be used to compute the best paths along the constrained topology. A given combination of calculation-type, metric-type, and constraints is known as a "Flexible Algorithm Definition". A router that sends such a set of TLVs also assigns a Flex-Algorithm value to the specified combination of calculation-type, metric-type, and constraints.

This document also specifies a way for a router to use IGPs to associate one or more SR Prefix-SIDs or SRv6 locators with a particular Flex-Algorithm. Each such Prefix-SID or SRv6 locator then represents a path that is computed according to the identified Flex-Algorithm.

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. Terminology

This section defines terms that are often used in this document.

Flexible Algorithm Definition (FAD) - the set consisting of (a) calculation-type, (b) metric-type, and (c) a set of constraints.

Flexible Algorithm - a numeric identifier in the range 128-255 that is associated via configuration with the Flexible-Algorithm Definition.

Local Flexible Algorithm Definition - Flexible Algorithm Definition defined locally on the node.

Remote Flexible Algorithm Definition - Flexible Algorithm Definition received from other nodes via IGP flooding.

Flexible Algorithm Participation - per application configuration state that expresses whether the node is participating in a particular Flexible Algorithm.

IGP Algorithm - value from the the "IGP Algorithm Types" registry defined under "Interior Gateway Protocol (IGP) Parameters" IANA registries. IGP Algorithms represents the triplet (Calculation Type, Metric, Constraints), where the second and third elements of the triple MAY be unspecified.

ABR - Area Border Router. In IS-IS terminology it is also known as L1/L2 router.

ASBR - Autonomous System Border Router.

4. Flexible Algorithm

Many possible constraints may be used to compute a path over a network. Some networks are deployed as multiple planes. A simple form of constraint may be to use a particular plane. A more sophisticated form of constraint can include some extended metric as described in [RFC8570]. Constraints which restrict paths to links with specific affinities or avoid links with specific affinities are also possible. Combinations of these are also possible.

To provide maximum flexibility, we want to provide a mechanism that allows a router to (a) identify a particular calculation-type, (b) metric-type, (c) describe a particular set of constraints, and (d) assign a numeric identifier, referred to as Flex-Algorithm, to the combination of that calculation-type, metric-type, and those constraints. We want the mapping between the Flex-Algorithm and its meaning to be flexible and defined by the user. As long as all routers in the domain have a common understanding as to what a particular Flex-Algorithm represents, the resulting routing computation is consistent and traffic is not subject to any looping.

The set consisting of (a) calculation-type, (b) metric-type, and (c) a set of constraints is referred to as a Flexible-Algorithm Definition.

Flexible-Algorithm is a numeric identifier in the range 128-255 that is associated via configuratin with the Flexible-Algorithm Definition.

IANA "IGP Algorithm Types" registry defines the set of values for IGP Algorithms. We propose to allocate the following values for Flex-Algorithms from this registry:

128-255 - Flex-Algorithms

5. Flexible Algorithm Definition Advertisement

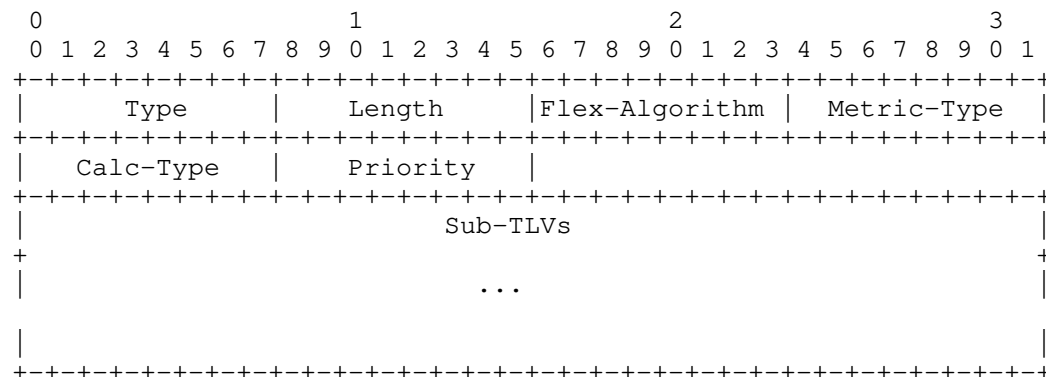
To guarantee the loop-free forwarding for paths computed for a particular Flex-Algorithm, all routers that (a) are configured to participate in a particular Flex-Algorithm, and (b) are in the same Flex-Algorithm definition advertisement scope MUST agree on the definition of the Flex-Algorithm.

5.1. IS-IS Flexible Algorithm Definition Sub-TLV

The IS-IS Flexible Algorithm Definition Sub-TLV (FAD Sub-TLV) is used to advertise the definition of the Flex-Algorithm.

The IS-IS FAD Sub-TLV is advertised as a Sub-TLV of the IS-IS Router Capability TLV-242 that is defined in [RFC7981].

IS-IS FAD Sub-TLV has the following format:



where:

Type: 26

Length: variable, dependent on the included Sub-TLVs

Flex-Algorithm: Single octet value between 128 and 255 inclusive.

Metric-Type: Type of metric to be used during the calculation.
Following values are defined:

0: IGP Metric

1: Min Unidirectional Link Delay as defined in [RFC8570], section 4.2, encoded as application specific link attribute as specified in [RFC8919] and Section 12 of this document.

2: Traffic Engineering Default Metric as defined in [RFC5305], section 3.7, encoded as application specific link attribute as specified in [RFC8919] and Section 12 of this document.

Calc-Type: value from 0 to 127 inclusive from the "IGP Algorithm Types" registry defined under "Interior Gateway Protocol (IGP) Parameters" IANA registries. IGP algorithms in the range of 0-127 have a defined triplet (Calculation Type, Metric, Constraints). When used to specify the Calc-Type in the FAD Sub-TLV, only the Calculation Type defined for the specified IGP Algorithm is used. The Metric/Constraints MUST NOT be inherited. If the required calculation type is Shortest Path First, the value 0 SHOULD appear in this field.

Priority: Value between 0 and 255 inclusive that specifies the priority of the advertisement.

Sub-TLVs - optional sub-TLVs.

The IS-IS FAD Sub-TLV MAY be advertised in an LSP of any number. IS-IS router MAY advertise more than one IS-IS FAD Sub-TLV for a given Flexible-Algorithm (see Section 6).

The IS-IS FAD Sub-TLV has an area scope. The Router Capability TLV in which the FAD Sub-TLV is present MUST have the S-bit clear.

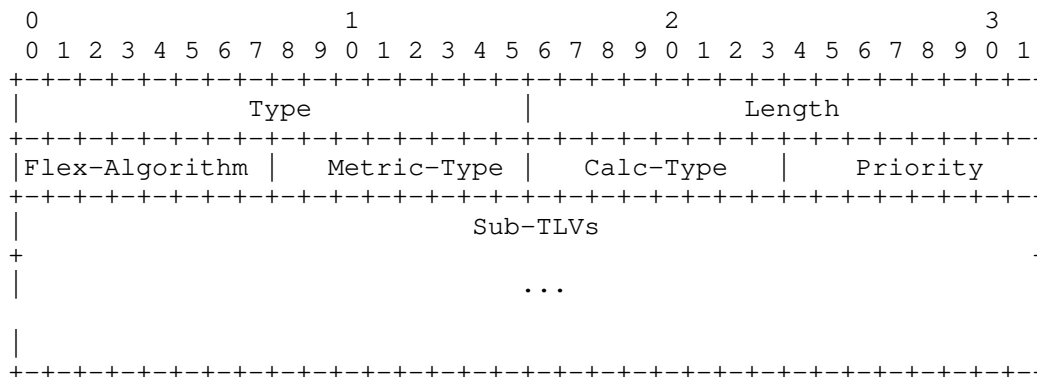
IS-IS L1/L2 router MAY be configured to re-generate the winning FAD from level 2, without any modification to it, to level 1 area. The re-generation of the FAD Sub-TLV from level 2 to level 1 is determined by the L1/L2 router, not by the originator of the FAD advertisement in the level 2. In such case, the re-generated FAD Sub-TLV will be advertised in the level 1 Router Capability TLV originated by the L1/L2 router.

L1/L2 router MUST NOT re-generate any FAD Sub-TLV from level 1 to level 2.

5.2. OSPF Flexible Algorithm Definition TLV

OSPF FAD TLV is advertised as a top-level TLV of the RI LSA that is defined in [RFC7770].

OSPF FAD TLV has the following format:



where:

Type: 16

Length: variable, dependent on the included Sub-TLVs

Flex-Algorithm:: Flex-Algorithm number. Value between 128 and 255 inclusive.

Metric-Type: Type of metric to be used during the calculation.
Following values are defined:

0: IGP Metric

1: Min Unidirectional Link Delay as defined in [RFC7471], section 4.2, encoded as application specific link attribute as specified in [RFC8920] and Section 12 of this document.

2: Traffic Engineering metric as defined in [RFC3630], section 2.5.5, encoded as application specific link attribute as specified in [RFC8920] and Section 12 of this document.

Calc-Type: as described in Section 5.1

Priority: as described in Section 5.1

Sub-TLVs - optional sub-TLVs.

When multiple OSPF FAD TLVs, for the same Flexible-Algorithm, are received from a given router, the receiver MUST use the first occurrence of the TLV in the Router Information LSA. If the OSPF FAD TLV, for the same Flex-Algorithm, appears in multiple Router Information LSAs that have different flooding scopes, the OSPF FAD TLV in the Router Information LSA with the area-scoped flooding scope MUST be used. If the OSPF FAD TLV, for the same algorithm, appears in multiple Router Information LSAs that have the same flooding scope, the OSPF FAD TLV in the Router Information (RI) LSA with the numerically smallest Instance ID MUST be used and subsequent instances of the OSPF FAD TLV MUST be ignored.

The RI LSA can be advertised at any of the defined opaque flooding scopes (link, area, or Autonomous System (AS)). For the purpose of OSPF FAD TLV advertisement, area-scoped flooding is REQUIRED. The Autonomous System flooding scope SHOULD NOT be used by default unless local configuration policy on the originating router indicates domain wide flooding.

5.3. Common Handling of Flexible Algorithm Definition TLV

This section describes the protocol-independent handling of the FAD TLV (OSPF) or FAD Sub-TLV (IS-IS). We will refer to it as FAD TLV in this section, even though in the case of IS-IS it is a Sub-TLV.

The value of the Flex-Algorithm MUST be between 128 and 255 inclusive. If it is not, the FAD TLV MUST be ignored.

Only a subset of the routers participating in the particular Flex-Algorithm need to advertise the definition of the Flex-Algorithm.

Every router, that is configured to participate in a particular Flex-Algorithm, MUST select the Flex-Algorithm definition based on the following ordered rules. This allows for the consistent Flex-Algorithm definition selection in cases where different routers advertise different definitions for a given Flex-Algorithm:

1. From the advertisements of the FAD in the area (including both locally generated advertisements and received advertisements) select the one(s) with the highest priority value.
2. If there are multiple advertisements of the FAD with the same highest priority, select the one that is originated from the router with the highest System-ID, in the case of IS-IS, or Router ID, in the case of OSPFv2 and OSPFv3. For IS-IS, the System-ID is

described in [ISO10589]. For OSPFv2 and OSPFv3, standard Router ID is described in [RFC2328] and [RFC5340] respectively.

A router that is not configured to participate in a particular Flex-Algorithm MUST ignore FAD Sub-TLVs advertisements for such Flex-Algorithm.

A router that is not participating in a particular Flex-Algorithm is allowed to advertise FAD for such Flex-Algorithm. Receiving routers MUST consider FAD advertisement regardless of the Flex-Algorithm participation of the FAD originator.

Any change in the Flex-Algorithm definition may result in temporary disruption of traffic that is forwarded based on such Flex-Algorithm paths. The impact is similar to any other event that requires network-wide convergence.

If a node is configured to participate in a particular Flexible-Algorithm, but there is no valid Flex-Algorithm definition available for it, or the selected Flex-Algorithm definition includes calculation-type, metric-type, constraint, flag, or Sub-TLV that is not supported by the node, it MUST stop participating in such Flexible-Algorithm. That implies that it MUST NOT announce participation for such Flexible-Algorithm as specified in Section 11 and it MUST remove any forwarding state associated with it.

Flex-Algorithm definition is topology independent. It applies to all topologies that a router participates in.

6. Sub-TLVs of IS-IS FAD Sub-TLV

One of the limitations of IS-IS [ISO10589] is that the length of a TLV/sub-TLV is limited to a maximum of 255 octets. For the FAD sub-TLV, there are a number of sub-sub-TLVs (defined below) which are supported. For a given Flex-Algorithm, it is possible that the total number of octets required to completely define a FAD exceeds the maximum length supported by a single FAD sub-TLV. In such cases, the FAD may be split into multiple such sub-TLVs and the content of the multiple FAD sub-TLVs combined to provide a complete FAD for the Flex-Algorithm. In such case, the fixed portion of the FAD (see Section 5.1) MUST be identical in all FAD sub-TLVs for a given Flex-Algorithm from a given IS. In case the fixed portion of such FAD Sub-TLVs differ, the values in the fixed portion in the FAD sub-TLV in the first occurrence in the lowest numbered LSP from a given IS MUST be used.

Any specification that introduces a new ISIS FAD sub-sub-TLV MUST specify whether the FAD sub-TLV may appear multiple times in the set

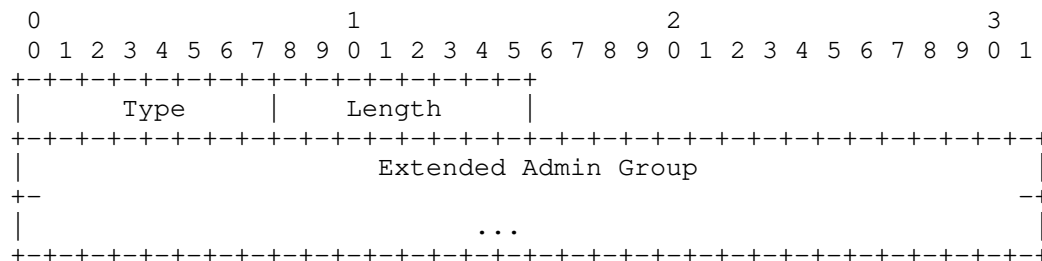
of FAD sub-TLVs for a given Flex-Algorithm from a given IS and how to handle them if multiple are allowed.

6.1. IS-IS Flexible Algorithm Exclude Admin Group Sub-TLV

The Flexible Algorithm definition can specify 'colors' that are used by the operator to exclude links during the Flex-Algorithm path computation.

The IS-IS Flexible Algorithm Exclude Admin Group Sub-TLV is used to advertise the exclude rule that is used during the Flex-Algorithm path calculation as specified in Section 13.

The IS-IS Flexible Algorithm Exclude Admin Group Sub-TLV (FAEAG Sub-TLV) is a Sub-TLV of the IS-IS FAD Sub-TLV. It has the following format:



where:

Type: 1

Length: variable, dependent on the size of the Extended Admin Group. MUST be a multiple of 4 octets.

Extended Administrative Group: Extended Administrative Group as defined in [RFC7308].

The IS-IS FAEAG Sub-TLV MUST NOT appear more than once in a single IS-IS FAD Sub-TLV. If it appears more than once, the IS-IS FAD Sub-TLV MUST be ignored by the receiver.

The IS-IS FAEAG Sub-TLV MUST NOT appear more than once in the set of FAD sub-TLVs for a given Flex-Algorithm from a given IS. If it appears more than once in such set, the IS-IS FAEAG Sub-TLV in the first occurrence in the lowest numbered LSP from a given IS MUST be used and any other occurrences MUST be ignored.

6.2. IS-IS Flexible Algorithm Include-Any Admin Group Sub-TLV

The Flexible Algorithm definition can specify 'colors' that are used by the operator to include links during the Flex-Algorithm path computation.

The IS-IS Flexible Algorithm Include-Any Admin Group Sub-TLV is used to advertise include-any rule that is used during the Flex-Algorithm path calculation as specified in Section 13.

The format of the IS-IS Flexible Algorithm Include-Any Admin Group Sub-TLV is identical to the format of the FAEAG Sub-TLV in Section 6.1.

The IS-IS Flexible Algorithm Include-Any Admin Group Sub-TLV Type is 2.

The IS-IS Flexible Algorithm Include-Any Admin Group Sub-TLV MUST NOT appear more than once in a single IS-IS FAD Sub-TLV. If it appears more than once, the IS-IS FAD Sub-TLV MUST be ignored by the receiver.

The IS-IS Flexible Algorithm Include-Any Admin Group Sub-TLV MUST NOT appear more than once in the set of FAD sub-TLVs for a given Flex-Algorithm from a given IS. If it appears more than once in such set, the IS-IS Flexible Algorithm Include-Any Admin Group Sub-TLV in the first occurrence in the lowest numbered LSP from a given IS MUST be used and any other occurrences MUST be ignored.

6.3. IS-IS Flexible Algorithm Include-All Admin Group Sub-TLV

The Flexible Algorithm definition can specify 'colors' that are used by the operator to include link during the Flex-Algorithm path computation.

The IS-IS Flexible Algorithm Include-All Admin Group Sub-TLV is used to advertise include-all rule that is used during the Flex-Algorithm path calculation as specified in Section 13.

The format of the IS-IS Flexible Algorithm Include-All Admin Group Sub-TLV is identical to the format of the FAEAG Sub-TLV in Section 6.1.

The IS-IS Flexible Algorithm Include-All Admin Group Sub-TLV Type is 3.

The IS-IS Flexible Algorithm Include-All Admin Group Sub-TLV MUST NOT appear more than once in a single IS-IS FAD Sub-TLV. If it appears

more than once, the IS-IS FAD Sub-TLV MUST be ignored by the receiver.

The IS-IS Flexible Algorithm Include-All Admin Group Sub-TLV MUST NOT appear more than once in the set of FAD sub-TLVs for a given Flex-Algorithm from a given IS. If it appears more than once in such set, the IS-IS Flexible Algorithm Include-All Admin Group Sub-TLV in the first occurrence in the lowest numbered LSP from a given IS MUST be used and any other occurrences MUST be ignored.

6.4. IS-IS Flexible Algorithm Definition Flags Sub-TLV

The IS-IS Flexible Algorithm Definition Flags Sub-TLV (FADF Sub-TLV) is a Sub-TLV of the IS-IS FAD Sub-TLV. It 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                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     ...                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

where:

Type: 4

Length: variable, non-zero number of octets of the Flags field

Flags:

```

      0 1 2 3 4 5 6 7...
+---+---+---+---+---+---+---+
|M| | | | | | |...
+---+---+---+---+---+---+---+

```

M-flag: when set, the Flex-Algorithm specific prefix metric MUST be used for inter-area and external prefix calculation. This flag is not applicable to prefixes advertised as SRv6 locators.

Bits are defined/sent starting with Bit 0 defined above. Additional bit definitions that may be defined in the future SHOULD be assigned in ascending bit order so as to minimize the number of bits that will need to be transmitted.

Undefined bits MUST be transmitted as 0.

Bits that are NOT transmitted MUST be treated as if they are set to 0 on receipt.

The IS-IS FADF Sub-TLV MUST NOT appear more than once in a single IS-IS FAD Sub-TLV. If it appears more than once, the IS-IS FAD Sub-TLV MUST be ignored by the receiver.

The IS-IS FADF Sub-TLV MUST NOT appear more than once in the set of FAD sub-TLVs for a given Flex-Algorithm from a given IS. If it appears more than once in such set, the IS-IS FADF Sub-TLV in the first occurrence in the lowest numbered LSP from a given IS MUST be used and any other occurrences MUST be ignored.

If the IS-IS FADF Sub-TLV is not present inside the IS-IS FAD Sub-TLV, all the bits are assumed to be set to 0.

If a node is configured to participate in a particular Flexible-Algorithm, but the selected Flex-Algorithm definition includes a bit in the IS-IS FADF Sub-TLV that is not supported by the node, it MUST stop participating in such Flexible-Algorithm.

New flag bits may be defined in the future. Implementations MUST check all advertised flag bits in the received IS-IS FADF Sub-TLV - not just the subset currently defined.

6.5. IS-IS Flexible Algorithm Exclude SRLG Sub-TLV

The Flexible Algorithm definition can specify Shared Risk Link Groups (SRLGs) that the operator wants to exclude during the Flex-Algorithm path computation.

The IS-IS Flexible Algorithm Exclude SRLG Sub-TLV (FAESRLG) is used to advertise the exclude rule that is used during the Flex-Algorithm path calculation as specified in Section 13.

The IS-IS FAESRLG Sub-TLV is a Sub-TLV of the IS-IS FAD Sub-TLV. It 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          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               Shared Risk Link Group Value                               |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               ...                               |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

where:

Type: 5

Length: variable, dependent on number of SRLG values. MUST be a multiple of 4 octets.

Shared Risk Link Group Value: SRLG value as defined in [RFC5307].

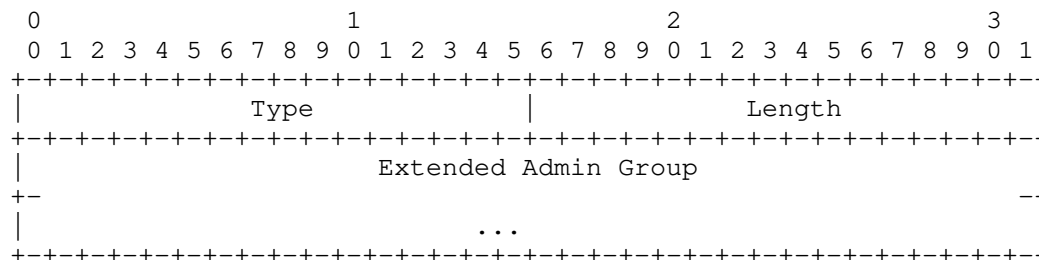
The IS-IS FAESRLG Sub-TLV MUST NOT appear more than once in a single IS-IS FAD Sub-TLV. If it appears more than once, the IS-IS FAD Sub-TLV MUST be ignored by the receiver.

The IS-IS FAESRLG Sub-TLV MAY appear more than once in the set of FAD sub-TLVs for a given Flex-Algorithm from a given IS. This may be necessary in cases where the total number of SRLG values which are specified cause the FAD sub-TLV to exceed the maximum length of a single FAD sub-TLV. In such case the receiver MUST use the union of all values across all IS-IS FAESRLG Sub-TLVs from such set.

7. Sub-TLVs of OSPF FAD TLV

7.1. OSPF Flexible Algorithm Exclude Admin Group Sub-TLV

The Flexible Algorithm Exclude Admin Group Sub-TLV (FAEAG Sub-TLV) is a Sub-TLV of the OSPF FAD TLV. It's usage is described in Section 6.1. It has the following format:



where:

Type: 1

Length: variable, dependent on the size of the Extended Admin Group. MUST be a multiple of 4 octets.

Extended Administrative Group: Extended Administrative Group as defined in [RFC7308].

The OSPF FAEAG Sub-TLV MUST NOT appear more than once in an OSPF FAD TLV. If it appears more than once, the OSPF FAD TLV MUST be ignored by the receiver.

7.2. OSPF Flexible Algorithm Include-Any Admin Group Sub-TLV

The usage of this Sub-TLVs is described in Section 6.2.

The format of the OSPF Flexible Algorithm Include-Any Admin Group Sub-TLV is identical to the format of the OSPF FAEAG Sub-TLV in Section 7.1.

The OSPF Flexible Algorithm Include-Any Admin Group Sub-TLV Type is 2.

The OSPF Flexible Algorithm Include-Any Admin Group Sub-TLV MUST NOT appear more than once in an OSPF FAD TLV. If it appears more than once, the OSPF FAD TLV MUST be ignored by the receiver.

7.3. OSPF Flexible Algorithm Include-All Admin Group Sub-TLV

The usage of this Sub-TLVs is described in Section 6.3.

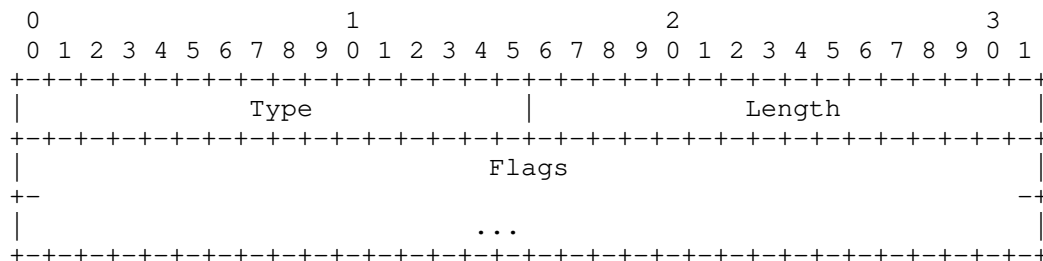
The format of the OSPF Flexible Algorithm Include-All Admin Group Sub-TLV is identical to the format of the OSPF FAEAG Sub-TLV in Section 7.1.

The OSPF Flexible Algorithm Include-All Admin Group Sub-TLV Type is 3.

The OSPF Flexible Algorithm Include-All Admin Group Sub-TLV MUST NOT appear more than once in an OSPF FAD TLV. If it appears more than once, the OSPF FAD TLV MUST be ignored by the receiver.

7.4. OSPF Flexible Algorithm Definition Flags Sub-TLV

The OSPF Flexible Algorithm Definition Flags Sub-TLV (FADF Sub-TLV) is a Sub-TLV of the OSPF FAD TLV. It has the following format:



where:

Type: 4

Length: variable, dependent on the size of the Flags field. MUST be a multiple of 4 octets.

Flags:

```

    0 1 2 3 4 5 6 7...
    +-+-+-+-+-+-+-+...
    |M| | |         ...
    +-+-+-+-+-+-+-+...

```

M-flag: when set, the Flex-Algorithm specific prefix and ASBR metric MUST be used for inter-area and external prefix calculation. This flag is not applicable to prefixes advertised as SRv6 locators.

Bits are defined/sent starting with Bit 0 defined above. Additional bit definitions that may be defined in the future SHOULD be assigned in ascending bit order so as to minimize the number of bits that will need to be transmitted.

Undefined bits MUST be transmitted as 0.

Bits that are NOT transmitted MUST be treated as if they are set to 0 on receipt.

The OSPF FADF Sub-TLV MUST NOT appear more than once in an OSPF FAD TLV. If it appears more than once, the OSPF FAD TLV MUST be ignored by the receiver.

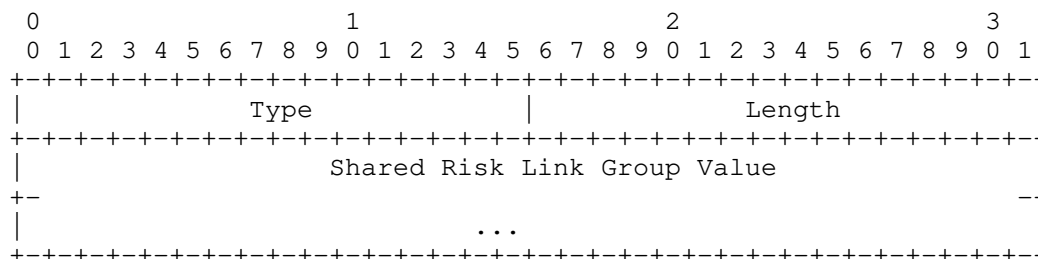
If the OSPF FADF Sub-TLV is not present inside the OSPF FAD TLV, all the bits are assumed to be set to 0.

If a node is configured to participate in a particular Flexible-Algorithm, but the selected Flex-Algorithm definition includes a bit in the OSPF FADF Sub-TLV that is not supported by the node, it MUST stop participating in such Flexible-Algorithm.

New flag bits may be defined in the future. Implementations MUST check all advertised flag bits in the received OSPF FADF Sub-TLV - not just the subset currently defined.

7.5. OSPF Flexible Algorithm Exclude SRLG Sub-TLV

The OSPF Flexible Algorithm Exclude SRLG Sub-TLV (FAESRLG Sub-TLV) is a Sub-TLV of the OSPF FAD TLV. Its usage is described in Section 6.5. It has the following format:



where:

Type: 5

Length: variable, dependent on the number of SRLGs. MUST be a multiple of 4 octets.

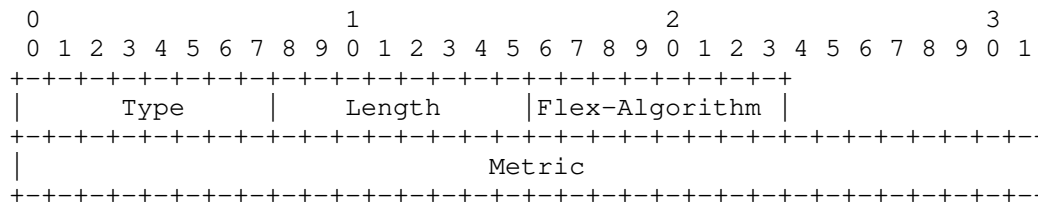
Shared Risk Link Group Value: SRLG value as defined in [RFC4203].

The OSPF FAESRLG Sub-TLV MUST NOT appear more than once in an OSPF FAD TLV. If it appears more than once, the OSPF FAD TLV MUST be ignored by the receiver.

8. IS-IS Flexible Algorithm Prefix Metric Sub-TLV

The IS-IS Flexible Algorithm Prefix Metric (FAPM) Sub-TLV supports the advertisement of a Flex-Algorithm specific prefix metric associated with a given prefix advertisement.

The IS-IS FAPM Sub-TLV is a sub-TLV of TLVs 135, 235, 236, and 237 and has the following format:



where:

Type: 6

Length: 5 octets

Flex-Algorithm: Single octet value between 128 and 255 inclusive.

Metric: 4 octets of metric information

The IS-IS FAPM Sub-TLV MAY appear multiple times in its parent TLV. If it appears more than once with the same Flex-Algorithm value, the first instance MUST be used and any subsequent instances MUST be ignored.

If a prefix is advertised with a Flex-Algorithm prefix metric larger than MAX_PATH_METRIC as defined in [RFC5305] this prefix MUST NOT be considered during the Flexible-Algorithm computation.

The usage of the Flex-Algorithm prefix metric is described in Section 13.

The IS-IS FAPM Sub-TLV MUST NOT be advertised as a sub-TLV of the IS-IS SRv6 Locator TLV [I-D.ietf-lsr-isis-srv6-extensions]. The IS-IS SRv6 Locator TLV includes the Algorithm and Metric fields which MUST be used instead. If the FAPM Sub-TLV is present as a sub-TLV of the IS-IS SRv6 Locator TLV in the received LSP, such FAPM Sub-TLV MUST be ignored.

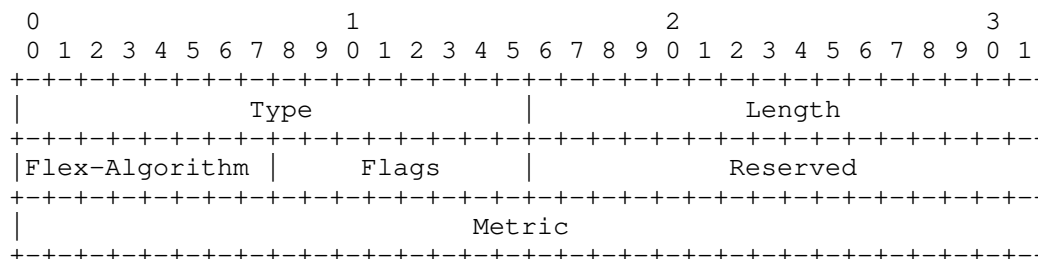
9. OSPF Flexible Algorithm Prefix Metric Sub-TLV

The OSPF Flexible Algorithm Prefix Metric (FAPM) Sub-TLV supports the advertisement of a Flex-Algorithm specific prefix metric associated with a given prefix advertisement.

The OSPF Flex-Algorithm Prefix Metric (FAPM) Sub-TLV is a Sub-TLV of the:

- OSPFv2 Extended Prefix TLV [RFC7684]
- Following OSPFv3 TLVs as defined in [RFC8362]:
 - Inter-Area Prefix TLV
 - External Prefix TLV

OSPF FAPM Sub-TLV has the following format:



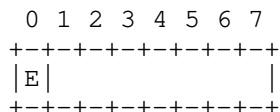
where:

Type: 3 for OSPFv2, 26 for OSPFv3

Length: 8 octets

Flex-Algorithm: Single octet value between 128 and 255 inclusive.

Flags: single octet value



E bit : position 0: The type of external metric. If bit is set, the metric specified is a Type 2 external metric. This bit is applicable only to OSPF External and NSSA external prefixes. This is semantically the same as E bit in section A.4.5 of [RFC2328] and section A.4.7 of [RFC5340] for OSPFv2 and OSPFv3 respectively.

Bits 1 through 7: MUST be cleared by sender and ignored by receiver.

Reserved: Must be set to 0, ignored at reception.

Metric: 4 octets of metric information

The OSPF FAPM Sub-TLV MAY appear multiple times in its parent TLV. If it appears more than once with the same Flex-Algorithm value, the first instance MUST be used and any subsequent instances MUST be ignored.

The usage of the Flex-Algorithm prefix metric is described in Section 13.

10. OSPF Flexible Algorithm ASBR Reachability Advertisement

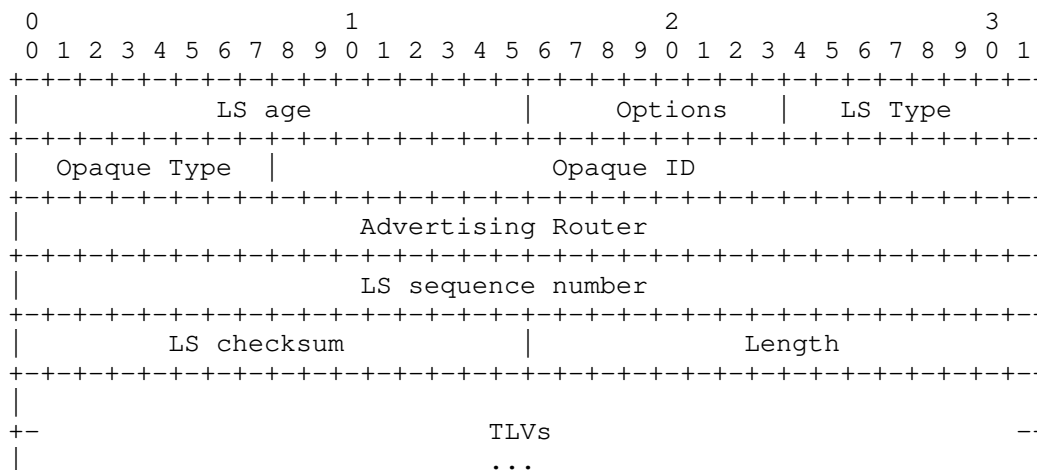
An OSPF ABR advertises the reachability of ASBRs in its attached areas to enable routers within those areas to perform route calculations for external prefixes advertised by the ASBRs. OSPF extensions for advertisement of Flex-Algorithm specific reachability and metric for ASBRs is similarly required for Flex-Algorithm external prefix computations as described further in Section 13.1.

10.1. OSPFv2 Extended Inter-Area ASBR LSA

The OSPFv2 Extended Inter-Area ASBR (EIA-ASBR) LSA is an OSPF Opaque LSA [RFC5250] that is used to advertise additional attributes related to the reachability of the OSPFv2 ASBR that is external to the area yet internal to the OSPF domain. Semantically, the OSPFv2 EIA-ASBR LSA is equivalent to the fixed format Type 4 Summary LSA [RFC2328]. Unlike the Type 4 Summary LSA, the LSID of the EIA-ASBR LSA does not carry the ASBR Router-ID - the ASBR Router-ID is carried in the body of the LSA. OSPFv2 EIA-ASBR LSA is advertised by an OSPFv2 ABR and its flooding is defined to be area-scoped only.

An OSPFv2 ABR generates the EIA-ASBR LSA for an ASBR when it is advertising the Type-4 Summary LSA for it and has the need for advertising additional attributes for that ASBR beyond what is conveyed in the fixed format Type-4 Summary LSA. An OSPFv2 ABR MUST NOT advertise the EIA-ASBR LSA for an ASBR for which it is not advertising the Type 4 Summary LSA. This ensures that the ABR does not generate the EIA-ASBR LSA for an ASBR to which it does not have reachability in the base OSPFv2 topology calculation. The OSPFv2 ABR SHOULD NOT advertise the EIA-ASBR LSA for an ASBR when it does not have additional attributes to advertise for that ASBR.

The OSPFv2 EIA-ASBR LSA has the following format:



The Opaque Type used by the OSPFv2 EIA-ASBR LSA is TBD (suggested value 11). The Opaque Type is used to differentiate the various types of OSPFv2 Opaque LSAs and is described in Section 3 of [RFC5250]. The LS Type MUST be 10, indicating that the Opaque LSA flooding scope is area-local [RFC5250]. The LSA Length field [RFC2328] represents the total length (in octets) of the Opaque LSA, including the LSA header and all TLVs (including padding).

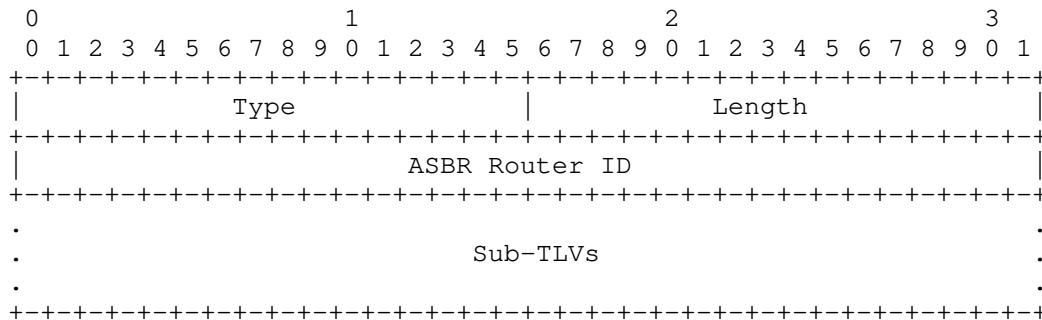
The Opaque ID field is an arbitrary value used to maintain multiple OSPFv2 EIA-ASBR LSAs. For OSPFv2 EIA-ASBR LSAs, the Opaque ID has no semantic significance other than to differentiate OSPFv2 EIA-ASBR LSAs originated by the same OSPFv2 ABR. If multiple OSPFv2 EIA-ASBR LSAs specify the same ASBR, the attributes from the Opaque LSA with the lowest Opaque ID SHOULD be used.

The format of the TLVs within the body of the OSPFv2 EIA-ASBR LSA is the same as the format used by the Traffic Engineering Extensions to OSPFv2 [RFC3630]. The variable TLV section consists of one or more nested TLV tuples. Nested TLVs are also referred to as sub-TLVs. The Length field defines the length of the value portion in octets (thus, a TLV with no value portion would have a length of 0). The TLV is padded to 4-octet alignment; padding is not included in the Length field (so a 3-octet value would have a length of 3, but the total size of the TLV would be 8 octets). Nested TLVs are also 32-bit aligned. For example, a 1-byte value would have the Length field set to 1, and 3 octets of padding would be added to the end of the value portion of the TLV. The padding is composed of zeros.

10.1.1. OSPFv2 Extended Inter-Area ASBR TLV

The OSPFv2 Extended Inter-Area ASBR (EIA-ASBR) TLV is a top-level TLV of the OSPFv2 EIA-ASBR LSA and is used to advertise additional attributes associated with the reachability of an ASBR.

The OSPFv2 EIA-ASBR TLV has the following format:



where:

Type: 1

Length: variable

ASBR Router ID: four octets carrying the OSPF Router ID of the ASBR whose information is being carried.

Sub-TLVs : variable

Only a single OSPFv2 EIA-ASBR TLV MUST be advertised in each OSPFv2 EIA-ASBR LSA and the receiver MUST ignore all instances of this TLV other than the first one in an LSA.

OSPFv2 EIA-ASBR TLV MUST be present inside an OSPFv2 EIA-ASBR LSA with at least a single sub-TLV included, otherwise the OSPFv2 EIA-ASBR LSA MUST be ignored by the receiver.

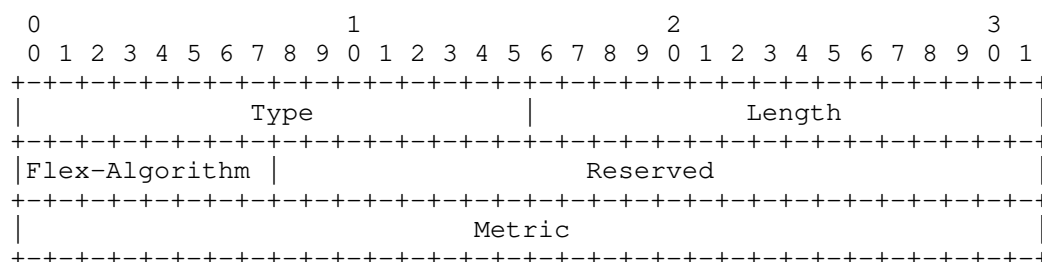
10.2. OSPF Flexible Algorithm ASBR Metric Sub-TLV

The OSPF Flexible Algorithm ASBR Metric (FAAM) Sub-TLV supports the advertisement of a Flex-Algorithm specific metric associated with a given ASBR reachability advertisement by an ABR.

The OSPF Flex-Algorithm ASBR Metric (FAAM) Sub-TLV is a Sub-TLV of the:

- OSPFv2 Extended Inter-Area ASBR TLV as defined in Section 10.1.1
- OSPFv3 Inter-Area-Router TLV defined in [RFC8362]

OSPF FAAM Sub-TLV has the following format:



where:

Type: 1 for OSPFv2, TBD (suggested value 30) for OSPFv3

Length: 8 octets

Flex-Algorithm: Single octet value between 128 and 255 inclusive.

Reserved: Must be set to 0, ignored at reception.

Metric: 4 octets of metric information

The OSPF FAAM Sub-TLV MAY appear multiple times in its parent TLV. If it appears more than once with the same Flex-Algorithm value, the first instance MUST be used and any subsequent instances MUST be ignored.

The advertisement of the ASBR reachability using the OSPF FAAM Sub-TLV inside the OSPFv2 EIA-ASBR LSA follows the section 12.4.3 of [RFC2328] and inside the OSPFv3 E-Inter-Area-Router LSA follows the section 4.8.5 of [RFC5340]. The reachability of the ASBR is evaluated in the context of the specific Flex-Algorithm.

The FAAM computed by the ABR will be equal to the metric to reach the ASBR for a given Flex-Algorithm in a source area or the cumulative metric via other ABR(s) when the ASBR is in a remote area. This is similar in nature to how the metric is set when the ASBR reachability metric is computed in the default algorithm for the metric in the OSPFv2 Type 4 ASBR Summary LSA and the OSPFv3 Inter-Area-Router LSA.

An OSPF ABR MUST NOT include the OSPF FAAM Sub-TLV with a specific Flex-Algorithm in its reachability advertisement for an ASBR between

areas unless that ASBR is reachable for it in the context of that specific Flex-Algorithm.

An OSPF ABR MUST include the OSPF FAAM Sub-TLVs as part of the ASBR reachability advertisement between areas for the Flex-Algorithm for which the winning FAD includes the M-flag and the ASBR is reachable in the context of that specific Flex-Algorithm.

OSPF routers MUST use the OSPF FAAM Sub-TLV to calculate the reachability of the ASBRs if the winning FAD for the specific Flex-Algorithm includes the M-flag. OSPF routers MUST NOT use the OSPF FAAM Sub-TLV to calculate the reachability of the ASBRs for the specific Flex-Algorithm if the winning FAD for such Flex-Algorithm does not include the M-flag. Instead, the OSPFv2 Type 4 Summary LSAs or the OSPFv3 Inter-Area-Router-LSAs MUST be used instead as specified in section 16.2 of [RFC2328] and section 4.8.5 of [RFC5340] for OSPFv2 and OSPFv3 respectively.

The processing of the new or changed OSPF FAAM Sub-TLV triggers the processing of the External routes similar to what is described in section 16.5 of the [RFC2328] for OSPFv2 and section 4.8.5 of [RFC5340] for OSPFv3 for the specific Flex-Algorithm. The External and NSSA External route calculation should be limited to Flex-Algorithm(s) for which the winning FAD(s) includes the M-flag.

Processing of the OSPF FAAM Sub-TLV does not require the existence of the equivalent OSPFv2 Type 4 Summary LSA or the OSPFv3 Inter-Area-Router-LSA that is advertised by the same ABR inside the area. When the OSPFv2 EIA-ASBR LSA or the OSPFv3 E-Inter-Area-Router-LSA are advertised along with the OSPF FAAM Sub-TLV by the ABR for a specific ASBR, it is expected that the same ABR would advertise the reachability of the same ASBR in the equivalent base LSAs - i.e., the OSPFv2 Type 4 Summary LSA or the OSPFv3 Inter-Area-Router-LSA. The presence of the base LSA is not mandatory for the usage of the extended LSA with the OSPF FAAM Sub-TLV. This means that the order in which these LSAs are received is not significant.

11. Advertisement of Node Participation in a Flex-Algorithm

When a router is configured to support a particular Flex-Algorithm, we say it is participating in that Flex-Algorithm.

Paths computed for a specific Flex-Algorithm MAY be used by various applications, each potentially using its own specific data plane for forwarding traffic over such paths. To guarantee the presence of the application specific forwarding state associated with a particular Flex-Algorithm, a router MUST advertise its participation for a particular Flex-Algorithm for each application specifically.

11.1. Advertisement of Node Participation for Segment Routing

[RFC8667], [RFC8665], and [RFC8666] (IGP Segment Routing extensions) describe how the SR-Algorithm is used to compute the IGP best path.

Routers advertise the support for the SR-Algorithm as a node capability as described in the above mentioned IGP Segment Routing extensions. To advertise participation for a particular Flex-Algorithm for Segment Routing, including both SR MPLS and SRv6, the Flex-Algorithm value MUST be advertised in the SR-Algorithm TLV (OSPF) or sub-TLV (IS-IS).

Segment Routing Flex-Algorithm participation advertisement is topology independent. When a router advertises participation in an SR-Algorithm, the participation applies to all topologies in which the advertising node participates.

11.2. Advertisement of Node Participation for Other Applications

This section describes considerations related to how other applications can advertise their participation in a specific Flex-Algorithm.

Application-specific Flex-Algorithm participation advertisements MAY be topology specific or MAY be topology independent, depending on the application itself.

Application-specific advertisement for Flex-Algorithm participation MUST be defined for each application and is outside of the scope of this document.

12. Advertisement of Link Attributes for Flex-Algorithm

Various link attributes may be used during the Flex-Algorithm path calculation. For example, include or exclude rules based on link affinities can be part of the Flex-Algorithm definition as defined in Section 6 and Section 7.

Application-specific link attributes, as specified in [RFC8919] or [RFC8920], that are to be used during Flex-Algorithm calculation MUST use the Application-Specific Link Attribute (ASLA) advertisements defined in [RFC8919] or [RFC8920], unless, in the case of IS-IS, the L-Flag is set in the ASLA advertisement. When the L-Flag is set, then legacy advertisements are to be used, subject to the procedures and constraints defined in [[RFC8919] Section 4.2 and Section 6.

The mandatory use of ASLA advertisements applies to link attributes specifically mentioned in this document (Min Unidirectional Link

Delay, TE Default Metric, Administrative Group, Extended Administrative Group and Shared Risk Link Group) and any other link attributes that may be used in support of Flex-Algorithm in the future.

A new Application Identifier Bit is defined to indicate that the ASLA advertisement is associated with the Flex-Algorithm application. This bit is set in the Standard Application Bit Mask (SABM) defined in [RFC8919] or [RFC8920]:

Bit-3: Flexible Algorithm (X-bit)

ASLA Admin Group Advertisements to be used by the Flexible Algorithm Application MAY use either the Administrative Group or Extended Administrative Group encodings. If the Administrative Group encoding is used, then the first 32 bits of the corresponding FAD sub-TLVs are mapped to the link attribute advertisements as specified in RFC 7308.

A receiver supporting this specification MUST accept both ASLA Administrative Group and Extended Administrative Group TLVs as defined in [RFC8919] or [RFC8920]. In the case of ISIS, if the L-Flag is set in ASLA advertisement, as defined in [RFC8919] Section 4.2, then the receiver MUST be able to accept both Administrative Group TLV as defined in [RFC5305] and Extended Administrative Group TLV as defined in [RFC7308].

13. Calculation of Flexible Algorithm Paths

A router MUST be configured to participate in a given Flex-Algorithm K and MUST select the FAD based on the rules defined in Section 5.3 before it can compute any path for that Flex-Algorithm.

No specific two way connectivity check is performed during the Flex-Algorithm path computation. The result of the existing, Flex-Algorithm agnostic, two way connectivity check is used during the Flex-Algorithm path computation.

As described in Section 11, participation for any particular Flex-Algorithm MUST be advertised on a per-application basis. Calculation of the paths for any particular Flex-Algorithm MUST be application specific.

The way applications handle nodes that do not participate in Flexible-Algorithm is application specific. If the application only wants to consider participating nodes during the Flex-Algorithm calculation, then when computing paths for a given Flex-Algorithm, all nodes that do not advertise participation for that Flex-Algorithm in their application-specific advertisements MUST be pruned from the

topology. Segment Routing, including both SR MPLS and SRv6, are applications that MUST use such pruning when computing Flex-Algorithm paths.

When computing the path for a given Flex-Algorithm, the metric-type that is part of the Flex-Algorithm definition (Section 5) MUST be used.

When computing the path for a given Flex-Algorithm, the calculation-type that is part of the Flex-Algorithm definition (Section 5) MUST be used.

Various link include or exclude rules can be part of the Flex-Algorithm definition. To refer to a particular bit within an AG or EAG we use the term 'color'.

Rules, in the order as specified below, MUST be used to prune links from the topology during the Flex-Algorithm computation.

For all links in the topology:

1. Check if any exclude AG rule is part of the Flex-Algorithm definition. If such exclude rule exists, check if any color that is part of the exclude rule is also set on the link. If such a color is set, the link MUST be pruned from the computation.
2. Check if any exclude SRLG rule is part of the Flex-Algorithm definition. If such exclude rule exists, check if the link is part of any SRLG that is also part of the SRLG exclude rule. If the link is part of such SRLG, the link MUST be pruned from the computation.
3. Check if any include-any AG rule is part of the Flex-Algorithm definition. If such include-any rule exists, check if any color that is part of the include-any rule is also set on the link. If no such color is set, the link MUST be pruned from the computation.
4. Check if any include-all AG rule is part of the Flex-Algorithm definition. If such include-all rule exists, check if all colors that are part of the include-all rule are also set on the link. If all such colors are not set on the link, the link MUST be pruned from the computation.
5. If the Flex-Algorithm definition uses other than IGP metric (Section 5), and such metric is not advertised for the particular link in a topology for which the computation is done, such link

MUST be pruned from the computation. A metric of value 0 MUST NOT be assumed in such case.

13.1. Multi-area and Multi-domain Considerations

Any IGP Shortest Path Tree calculation is limited to a single area. This applies to Flex-Algorithm calculations as well. Given that the computing router does not have visibility of the topology of the next areas or domain, the Flex-Algorithm specific path to an inter-area or inter-domain prefix will be computed for the local area only. The egress L1/L2 router (ABR in OSPF), or ASBR for inter-domain case, will be selected based on the best path for the given Flex-Algorithm in the local area and such egress ABR or ASBR router will be responsible to compute the best Flex-Algorithm specific path over the next area or domain. This may produce an end-to-end path, which is sub-optimal based on Flex-Algorithm constraints. In cases where the ABR or ASBR has no reachability to a prefix for a given Flex-Algorithm in the next area or domain, the traffic may be dropped by the ABR/ASBR.

To allow the optimal end-to-end path for an inter-area or inter-domain prefix for any Flex-Algorithm to be computed, the FAPM has been defined in Section 8 and Section 9. For external route calculation for prefixes originated by ASBRs in remote areas in OSPF, the FAAM has been defined in Section 10.2 for the ABR to indicate its ASBR reachability along with the metric for the specific Flex-Algorithm.

If the FAD selected based on the rules defined in Section 5.3 includes the M-flag, an ABR or ASBR MUST include the FAPM (Section 8, Section 9) when advertising the prefix, that is reachable in a given Flex-Algorithm, between areas or domains. Such metric will be equal to the metric to reach the prefix for that Flex-Algorithm in its source area or domain. This is similar in nature to how the metric is set when prefixes are advertised between areas or domains for the default algorithm. When a prefix is unreachable in its source area or domain in a specific Flex-Algorithm, then an ABR or ASBR MUST NOT include the FAPM for that Flex-Algorithm when advertising the prefix between areas or domains.

If the FAD selected based on the rules defined in Section 5.3 includes the M-flag, the FAPM MUST be used during the calculation of prefix reachability for the inter-area and external prefixes. If the FAPM for the Flex-Algorithm is not advertised with the inter-area or external prefix reachability advertisement, the prefix MUST be considered as unreachable for that Flex-Algorithm. Similarly in the case of OSPF, for ASBRs in remote areas, if the FAAM is not advertised by the local ABR(s), the ASBR MUST be considered as

unreachable for that Flex-Algorithm and the external prefix advertisements from such an ASBR are not considered for that Flex-Algorithm.

Flex-Algorithm prefix metrics and the OSPF Flex-Algorithm ASBR metrics MUST NOT be used during the Flex-Algorithm computation unless the FAD selected based on the rules defined in Section 5.3 includes the M-Flag, as described in (Section 6.4 or Section 7.4).

In the case of OSPF, when calculating external routes in a Flex-Algorithm (with FAD selected includes the M-Flag) where the advertising ASBR is in a remote area, the metric will be the sum of the following:

- o the FAPM for that Flex-Algorithm advertised with the external route by the ASBR
- o the metric to reach the ASBR for that Flex-Algorithm from the local ABR i.e., the FAAM for that Flex-Algorithm advertised by the ABR in the local area for that ASBR
- o the Flex-Algorithm specific metric to reach the local ABR

This is similar in nature to how the metric is calculated for routes learned from remote ASBRs in the default algorithm using the OSPFv2 Type 4 ASBR Summary LSA and the OSPFv3 Inter-Area-Router LSA.

If the FAD selected based on the rules defined in Section 5.3 does not includes the M-flag, then the IGP metrics associated with the prefix reachability advertisements used by the base IS-IS and OSPF protocol MUST be used for the Flex-Algorithm route computation. Similarly, in the case of external route calculations in OSPF, the ASBR reachability is determined based on the base OSPFv2 Type 4 Summary LSA and the OSPFv3 Inter-Area-Router LSA.

It is NOT RECOMMENDED to use the Flex-Algorithm for inter-area or inter-domain prefix reachability without the M-flag set. The reason is that without the explicit Flex-Algorithm Prefix Metric advertisement (and the Flex-Algorithm ASBR metric advertisement in the case of OSPF external route calculation), it is not possible to conclude whether the ABR or ASBR has reachability to the inter-area or inter-domain prefix for a given Flex-Algorithm in the next area or domain. Sending the Flex-Algorithm traffic for such prefix towards the ABR or ASBR may result in traffic looping or black-holing.

During the route computation, it is possible for the Flex-Algorithm specific metric to exceed the maximum value that can be stored in an unsigned 32-bit variable. In such scenarios, the value MUST be

considered to be of value 4,294,967,295 during the computation and advertised as such.

The FAPM MUST NOT be advertised with IS-IS L1 or L2 intra-area, OSPFv2 intra-area, or OSPFv3 intra-area routes. If the FAPM is advertised for these route-types, it MUST be ignored during the prefix reachability calculation.

The M-flag in FAD is not applicable to prefixes advertised as SRv6 locators. The IS-IS SRv6 Locator TLV [I-D.ietf-lsr-isis-srv6-extensions] includes the Algorithm and Metric fields. When the SRv6 Locator is advertised between areas or domains, the metric field in the Locator TLV of IS-IS MUST be used irrespective of the M-flag in the FAD advertisement.

OSPF external and NSSA external prefix advertisements MAY include a non-zero forwarding address in the prefix advertisements in the base protocol. In such a scenario, the Flex-Algorithm specific reachability of the external prefix is determined by Flex-Algorithm specific reachability of the forwarding address.

In OSPF, the procedures for translation of NSSA external prefix advertisements into external prefix advertisements performed by an NSSA ABR [RFC3101] remain unchanged for Flex-Algorithm. An NSSA translator MUST include the OSPF FAPM Sub-TLVs for all Flex-Algorithms that are in the original NSSA external prefix advertisement from the NSSA ASBR in the translated external prefix advertisement generated by it regardless of its participation in those Flex-Algorithms or its having reachability to the NSSA ASBR in those Flex-Algorithms.

An area could become partitioned from the perspective of the Flex-Algorithm due to the constraints and/or metric being used for it, while maintaining the continuity in the algorithm 0. When that happens, some destinations inside that area could become unreachable in that Flex-Algorithm. These destinations will not be able to use an inter-area path. This is the consequence of the fact that the inter-area prefix reachability advertisement would not be available for these intra-area destinations within the area. It is RECOMMENDED to avoid such partitioning by providing enough redundancy inside the area for each Flex-Algorithm being used.

14. Flex-Algorithm and Forwarding Plane

This section describes how Flex-Algorithm paths are used in forwarding.

14.1. Segment Routing MPLS Forwarding for Flex-Algorithm

This section describes how Flex-Algorithm paths are used with SR MPLS forwarding.

Prefix SID advertisements include an SR-Algorithm value and, as such, are associated with the specified SR-Algorithm. Prefix-SIDs are also associated with a specific topology which is inherited from the associated prefix reachability advertisement. When the algorithm value advertised is a Flex-Algorithm value, the Prefix SID is associated with paths calculated using that Flex-Algorithm in the associated topology.

A Flex-Algorithm path **MUST** be installed in the MPLS forwarding plane using the MPLS label that corresponds to the Prefix-SID that was advertised for that Flex-algorithm. If the Prefix SID for a given Flex-algorithm is not known, the Flex-Algorithm specific path cannot be installed in the MPLS forwarding plane.

Traffic that is supposed to be routed via Flex-Algorithm specific paths, **MUST** be dropped when there are no such paths available.

Loop Free Alternate (LFA) paths for a given Flex-Algorithm **MUST** be computed using the same constraints as the calculation of the primary paths for that Flex-Algorithm. LFA paths **MUST** only use Prefix-SIDs advertised specifically for the given algorithm. LFA paths **MUST NOT** use an Adjacency-SID that belongs to a link that has been pruned from the Flex-Algorithm computation.

If LFA protection is being used to protect a given Flex-Algorithm paths, all routers in the area participating in the given Flex-Algorithm **SHOULD** advertise at least one Flex-Algorithm specific Node-SID. These Node-SIDs are used to steer traffic over the LFA computed backup path.

14.2. SRv6 Forwarding for Flex-Algorithm

This section describes how Flex-Algorithm paths are used with SRv6 forwarding.

In SRv6 a node is provisioned with topology/algorithm specific locators for each of the topology/algorithm pairs supported by that node. Each locator is an aggregate prefix for all SIDs provisioned on that node which have the matching topology/algorithm.

The SRv6 locator advertisement in IS-IS [I-D.ietf-lsr-isis-srv6-extensions] includes the MTID value that associates the locator with a specific topology. SRv6 locator

advertisements also includes an Algorithm value that explicitly associates the locator with a specific algorithm. When the algorithm value advertised with a locator represents a Flex-Algorithm, the paths to the locator prefix MUST be calculated using the specified Flex-Algorithm in the associated topology.

Forwarding entries for the locator prefixes advertised in IS-IS MUST be installed in the forwarding plane of the receiving SRv6 capable routers when the associated topology/algorithm is participating in them. Forwarding entries for locators associated with Flex-Algorithms in which the node is not participating MUST NOT be installed in the forwarding plane.

When the locator is associated with a Flex-Algorithm, LFA paths to the locator prefix MUST be calculated using such Flex-Algorithm in the associated topology, to guarantee that they follow the same constraints as the calculation of the primary paths. LFA paths MUST only use SRv6 SIDs advertised specifically for the given Flex-Algorithm.

If LFA protection is being used to protect locators associated with a given Flex-Algorithm, all routers in the area participating in the given Flex-Algorithm SHOULD advertise at least one Flex-Algorithm specific locator and END SID per node and one END.X SID for every link that has not been pruned from such Flex-Algorithm computation. These locators and SIDs are used to steer traffic over the LFA-computed backup path.

14.3. Other Applications' Forwarding for Flex-Algorithm

Any application that wants to use Flex-Algorithm specific forwarding needs to install some form of Flex-Algorithm specific forwarding entries.

Application-specific forwarding for Flex-Algorithm MUST be defined for each application and is outside of the scope of this document.

15. Operational Considerations

15.1. Inter-area Considerations

The scope of the Flex-Algorithm computation is an area, so is the scope of the FAD. In IS-IS, the Router Capability TLV in which the FAD Sub-TLV is advertised MUST have the S-bit clear, which prevents it to be flooded outside of the level in which it was originated. Even though in OSPF the FAD Sub-TLV can be flooded in an RI LSA that has AS flooding scope, the FAD selection is performed for each individual area in which it is being used.

There is no requirement for the FAD for a particular Flex-Algorithm to be identical in all areas in the network. For example, traffic for the same Flex-Algorithm may be optimized for minimal delay (e.g., using delay metric) in one area or level, while being optimized for available bandwidth (e.g., using IGP metric) in another area or level.

As described in Section 5.1, IS-IS allows the re-generation of the winning FAD from level 2, without any modification to it, into a level 1 area. This allows the operator to configure the FAD in one or multiple routers in the level 2, without the need to repeat the same task in each level 1 area, if the intent is to have the same FAD for the particular Flex-Algorithm across all levels. This can similarly be achieved in OSPF by using the AS flooding scope of the RI LSA in which the FAD Sub-TLV for the particular Flex-Algorithm is advertised.

Re-generation of FAD from a level 1 area to the level 2 area is not supported in IS-IS, so if the intent is to regenerate the FAD between IS-IS levels, the FAD MUST be defined on router(s) that are in level 2. In OSPF, the FAD definition can be done in any area and be propagated to all routers in the OSPF routing domain by using the AS flooding scope of the RI LSA.

15.2. Usage of SRLG Exclude Rule with Flex-Algorithm

There are two different ways in which SRLG information can be used with Flex-Algorithm:

- In a context of a single Flex-Algorithm, it can be used for computation of backup paths, as described in [I-D.ietf-rtgwg-segment-routing-ti-lfa]. This usage does not require association of any specific SRLG constraint with the given Flex-Algorithm definition.

- In the context of multiple Flex-Algorithms, it can be used for creating disjoint sets of paths by pruning the links belonging to a specific SRLG from the topology on which a specific Flex-Algorithm computes its paths. This usage:

 - Facilitates the usage of already deployed SRLG configurations for setup of disjoint paths between two or more Flex-Algorithms.

 - Requires explicit association of a given Flex-Algorithm with a specific set of SRLG constraints as defined in Section 6.5 and Section 7.5.

The two usages mentioned above are orthogonal.

15.3. Max-metric consideration

Both IS-IS and OSPF have a mechanism to set the IGP metric on a link to a value that would make the link either non-reachable or to serve as the link of last resort. Similar functionality would be needed for the Min Unidirectional Link Delay and TE metric, as these can be used to compute Flex-Algorithm paths.

The link can be made un-reachable for all Flex-Algorithms that use Min Unidirectional Link Delay as metric, as described in Section 5.1, by removing the Flex-Algorithm ASLA Min Unidirectional Link Delay advertisement for the link. The link can be made the link of last resort by setting the delay value in the Flex-Algorithm ASLA delay advertisement for the link to the value of 16,777,215 ($2^{24} - 1$).

The link can be made un-reachable for all Flex-Algorithms that use TE metric, as described in Section 5.1, by removing the Flex-Algorithm ASLA TE metric advertisement for the link. The link can be made the link of last resort by setting the TE metric value in the Flex-Algorithm ASLA delay advertisement for the link to the value of ($2^{24} - 1$) in IS-IS and ($2^{32} - 1$) in OSPF.

16. Backward Compatibility

This extension brings no new backward compatibility issues. IS-IS, OSPFv2 and OSPFv3 all have well defined handling of unrecognized TLVs and sub-TLVs that allows the introduction of the new extensions, similar to those defined here, without introducing any interoperability issues.

17. Security Considerations

This draft adds two new ways to disrupt IGP networks:

An attacker can hijack a particular Flex-Algorithm by advertising a FAD with a priority of 255 (or any priority higher than that of the legitimate nodes).

An attacker could make it look like a router supports a particular Flex-Algorithm when it actually doesn't, or vice versa.

Both of these attacks can be addressed by the existing security extensions as described in [RFC5304] and [RFC5310] for IS-IS, in [RFC2328] and [RFC7474] for OSPFv2, and in [RFC5340] and [RFC4552] for OSPFv3.

18. IANA Considerations

18.1. IGP IANA Considerations

18.1.1. IGP Algorithm Types Registry

This document makes the following registrations in the "IGP Algorithm Types" registry:

Type: 128-255.

Description: Flexible Algorithms.

Reference: This document (Section 4).

18.1.2. IGP Metric-Type Registry

IANA is requested to set up a registry called "IGP Metric-Type Registry" under an "Interior Gateway Protocol (IGP) Parameters" IANA registries. The registration policy for this registry is "Standards Action" ([RFC8126] and [RFC7120]).

Values in this registry come from the range 0-255.

This document registers following values in the "IGP Metric-Type Registry":

Type: 0

Description: IGP metric

Reference: This document (Section 5.1)

Type: 1

Description: Min Unidirectional Link Delay as defined in [RFC8570], section 4.2, and [RFC7471], section 4.2.

Reference: This document (Section 5.1)

Type: 2

Description: Traffic Engineering Default Metric as defined in [RFC5305], section 3.7, and Traffic engineering metric as defined in [RFC3630], section 2.5.5

Reference: This document (Section 5.1)

18.2. Flexible Algorithm Definition Flags Registry

IANA is requested to set up a registry called "IS-IS Flexible Algorithm Definition Flags Registry" under an "Interior Gateway Protocol (IGP) Parameters" IANA registries. The registration policy for this registry is "Standards Action" ([RFC8126] and [RFC7120]).

This document defines the following single bit in Flexible Algorithm Definition Flags registry:

Bit #	Name
-----	-----
0	Prefix Metric Flag (M-flag)

Reference: This document (Section 6.4, Section 7.4).

18.3. IS-IS IANA Considerations

18.3.1. Sub TLVs for Type 242

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

Type: 26.

Description: Flexible Algorithm Definition.

Reference: This document (Section 5.1).

18.3.2. Sub TLVs for for TLVs 135, 235, 236, and 237

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

Type: 6

Description: Flexible Algorithm Prefix Metric.

Reference: This document (Section 8).

18.3.3. Sub-Sub-TLVs for Flexible Algorithm Definition Sub-TLV

This document creates the following Sub-Sub-TLV Registry:

Registry: Sub-Sub-TLVs for Flexible Algorithm Definition Sub-TLV

Registration Procedure: Expert review

Reference: This document (Section 5.1)

This document defines the following Sub-Sub-TLVs in the "Sub-Sub-TLVs for Flexible Algorithm Definition Sub-TLV" registry:

Type: 1

Description: Flexible Algorithm Exclude Admin Group

Reference: This document (Section 6.1).

Type: 2

Description: Flexible Algorithm Include-Any Admin Group

Reference: This document (Section 6.2).

Type: 3

Description: Flexible Algorithm Include-All Admin Group

Reference: This document (Section 6.3).

Type: 4

Description: Flexible Algorithm Definition Flags

Reference: This document (Section 6.4).

Type: 5

Description: Flexible Algorithm Exclude SRLG

Reference: This document (Section 6.5).

18.4. OSPF IANA Considerations

18.4.1. OSPF Router Information (RI) TLVs Registry

This specification updates the OSPF Router Information (RI) TLVs Registry.

Type: 16

Description: Flexible Algorithm Definition TLV.

Reference: This document (Section 5.2).

18.4.2. OSPFv2 Extended Prefix TLV Sub-TLVs

This document makes the following registrations in the "OSPFv2 Extended Prefix TLV Sub-TLVs" registry.

Type: 3

Description: Flexible Algorithm Prefix Metric.

Reference: This document (Section 9).

18.4.3. OSPFv3 Extended-LSA Sub-TLVs

This document makes the following registrations in the "OSPFv3 Extended-LSA Sub-TLVs" registry.

Type: 26

Description: Flexible Algorithm Prefix Metric.

Reference: This document (Section 9).

Type: TBD (suggested value 30)

Description: OSPF Flexible Algorithm ASBR Metric Sub-TLV

Reference: This document (Section 10.2).

18.4.4. OSPF Flex-Algorithm Prefix Metric Bits

This specification requests creation of "OSPF Flex-Algorithm Prefix Metric Bits" registry under the OSPF Parameters Registry with the following initial values.

Bit Number: 0

Description: E bit - External Type

Reference: this document.

The bits 1-7 are unassigned and the registration procedure to be followed for this registry is IETF Review.

18.4.5. OSPF Opaque LSA Option Types

This document makes the following registrations in the "OSPF Opaque LSA Option Types" registry.

Value: TBD (suggested value 11)

Description: OSPFv2 Extended Inter-Area ASBR LSA

Reference: This document (Section 10.1).

18.4.6. OSPFv2 Extended Inter-Area ASBR TLVs

This specification requests creation of "OSPFv2 Extended Inter-Area ASBR TLVs" registry under the OSPFv2 Parameters Registry with the following initial values.

Value: 1

Description : Extended Inter-Area ASBR TLV

Reference: this document

The values 2 to 32767 are unassigned, values 32768 to 33023 are reserved for experimental use while the values 0 and 33024 to 65535 are reserved. The registration procedure to be followed for this registry is IETF Review or IESG Approval.

18.4.7. OSPFv2 Inter-Area ASBR Sub-TLVs

This specification requests creation of "OSPFv2 Extended Inter-Area ASBR Sub-TLVs" registry under the OSPFv2 Parameters Registry with the following initial values.

Value: 1

Description : OSPF Flexible Algorithm ASBR Metric Sub-TLV

Reference: this document

The values 2 to 32767 are unassigned, values 32768 to 33023 are reserved for experimental use while the values 0 and 33024 to 65535 are reserved. The registration procedure to be followed for this registry is IETF Review or IESG Approval.

18.4.8. OSPF Flexible Algorithm Definition TLV Sub-TLV Registry

This document creates the following registry:

Registry: OSPF Flexible Algorithm Definition TLV sub-TLV

Registration Procedure: Expert review

Reference: This document (Section 5.2)

The "OSPF Flexible Algorithm Definition TLV sub-TLV" registry will define sub-TLVs at any level of nesting for the Flexible Algorithm TLV and should be added to the "Open Shortest Path First (OSPF) Parameters" registries group. New values can be allocated via IETF Review or IESG Approval.

This document registers following Sub-TLVs in the "TLVs for Flexible Algorithm Definition TLV" registry:

Type: 1

Description: Flexible Algorithm Exclude Admin Group

Reference: This document (Section 7.1).

Type: 2

Description: Flexible Algorithm Include-Any Admin Group

Reference: This document (Section 7.2).

Type: 3

Description: Flexible Algorithm Include-All Admin Group

Reference: This document (Section 7.3).

Type: 4

Description: Flexible Algorithm Definition Flags

Reference: This document (Section 7.4).

Type: 5

Description: Flexible Algorithm Exclude SRLG

Reference: This document (Section 7.5).

Types in the range 32768-33023 are for experimental use; these will not be registered with IANA, and MUST NOT be mentioned by RFCs.

Types in the range 33024-65535 are not to be assigned at this time. Before any assignments can be made in the 33024-65535 range, there MUST be an IETF specification that specifies IANA Considerations that covers the range being assigned.

18.4.9. Link Attribute Applications Registry

This document registers following bit in the Link Attribute Applications Registry:

Bit-3

Description: Flexible Algorithm (X-bit)

Reference: This document (Section 12).

19. Acknowledgements

This draft, among other things, is also addressing the problem that the [I-D.gulkohegde-routing-planes-using-sr] was trying to solve. All authors of that draft agreed to join this draft.

Thanks to Eric Rosen, Tony Przygienda, William Britto A J, Gunter Van De Velde, Dirk Goethals, Manju Sivaji and, Baalajee S for their detailed review and excellent comments.

Thanks to Cengiz Halit for his review and feedback during initial phase of the solution definition.

Thanks to Kenji Kumaki for his comments.

Thanks to Acee Lindem for editorial comments.

20. References

20.1. Normative References

[I-D.ietf-lsr-isis-srv6-extensions]

Psenak, P., Filss, C., Bashandy, A., Decraene, B., and Z. Hu, "IS-IS Extensions to Support Segment Routing over IPv6 Dataplane", draft-ietf-lsr-isis-srv6-extensions-18 (work in progress), October 2021.

[ISO10589]

International Organization for Standardization, "Intermediate system to Intermediate system intra-domain routing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473)", 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>>.
- [RFC4203] Kompella, K., Ed. and Y. Rekhter, Ed., "OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4203, DOI 10.17487/RFC4203, October 2005, <<https://www.rfc-editor.org/info/rfc4203>>.
- [RFC5250] Berger, L., Bryskin, I., Zinin, A., and R. Coltun, "The OSPF Opaque LSA Option", RFC 5250, DOI 10.17487/RFC5250, July 2008, <<https://www.rfc-editor.org/info/rfc5250>>.
- [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>>.
- [RFC7308] Osborne, E., "Extended Administrative Groups in MPLS Traffic Engineering (MPLS-TE)", RFC 7308, DOI 10.17487/RFC7308, July 2014, <<https://www.rfc-editor.org/info/rfc7308>>.
- [RFC7684] Psenak, P., Gredler, H., Shakir, R., Henderickx, W., Tantsura, J., and A. Lindem, "OSPFv2 Prefix/Link Attribute Advertisement", RFC 7684, DOI 10.17487/RFC7684, November 2015, <<https://www.rfc-editor.org/info/rfc7684>>.
- [RFC7770] Lindem, A., Ed., Shen, N., Vasseur, JP., Aggarwal, R., and S. Shaffer, "Extensions to OSPF for Advertising Optional Router Capabilities", RFC 7770, DOI 10.17487/RFC7770, February 2016, <<https://www.rfc-editor.org/info/rfc7770>>.
- [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>>.
- [RFC8362] Lindem, A., Roy, A., Goethals, D., Reddy Vallem, V., and F. Baker, "OSPFv3 Link State Advertisement (LSA) Extensibility", RFC 8362, DOI 10.17487/RFC8362, April 2018, <<https://www.rfc-editor.org/info/rfc8362>>.

- [RFC8665] Psenak, P., Ed., Previdi, S., Ed., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", RFC 8665, DOI 10.17487/RFC8665, December 2019, <<https://www.rfc-editor.org/info/rfc8665>>.
- [RFC8666] Psenak, P., Ed. and S. Previdi, Ed., "OSPFv3 Extensions for Segment Routing", RFC 8666, DOI 10.17487/RFC8666, December 2019, <<https://www.rfc-editor.org/info/rfc8666>>.
- [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>>.
- [RFC8919] Ginsberg, L., Psenak, P., Previdi, S., Henderickx, W., and J. Drake, "IS-IS Application-Specific Link Attributes", RFC 8919, DOI 10.17487/RFC8919, October 2020, <<https://www.rfc-editor.org/info/rfc8919>>.
- [RFC8920] Psenak, P., Ed., Ginsberg, L., Henderickx, W., Tantsura, J., and J. Drake, "OSPF Application-Specific Link Attributes", RFC 8920, DOI 10.17487/RFC8920, October 2020, <<https://www.rfc-editor.org/info/rfc8920>>.

20.2. Informative References

- [I-D.gulkohegde-routing-planes-using-sr] Hegde, S. and A. Gulko, "Separating Routing Planes using Segment Routing", draft-gulkohegde-routing-planes-using-sr-00 (work in progress), March 2017.
- [I-D.ietf-rtgwg-segment-routing-ti-lfa] Litkowski, S., Bashandy, A., Filsfils, C., Francois, P., Decraene, B., and D. Voyer, "Topology Independent Fast Reroute using Segment Routing", draft-ietf-rtgwg-segment-routing-ti-lfa-08 (work in progress), January 2022.
- [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <<https://www.rfc-editor.org/info/rfc2328>>.
- [RFC3101] Murphy, P., "The OSPF Not-So-Stubby Area (NSSA) Option", RFC 3101, DOI 10.17487/RFC3101, January 2003, <<https://www.rfc-editor.org/info/rfc3101>>.

- [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, DOI 10.17487/RFC3630, September 2003, <<https://www.rfc-editor.org/info/rfc3630>>.
- [RFC3906] Shen, N. and H. Smit, "Calculating Interior Gateway Protocol (IGP) Routes Over Traffic Engineering Tunnels", RFC 3906, DOI 10.17487/RFC3906, October 2004, <<https://www.rfc-editor.org/info/rfc3906>>.
- [RFC4552] Gupta, M. and N. Melam, "Authentication/Confidentiality for OSPFv3", RFC 4552, DOI 10.17487/RFC4552, June 2006, <<https://www.rfc-editor.org/info/rfc4552>>.
- [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>>.
- [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>>.
- [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <<https://www.rfc-editor.org/info/rfc5340>>.
- [RFC7120] Cotton, M., "Early IANA Allocation of Standards Track Code Points", BCP 100, RFC 7120, DOI 10.17487/RFC7120, January 2014, <<https://www.rfc-editor.org/info/rfc7120>>.
- [RFC7471] Giacalone, S., Ward, D., Drake, J., Atlas, A., and S. Previdi, "OSPF Traffic Engineering (TE) Metric Extensions", RFC 7471, DOI 10.17487/RFC7471, March 2015, <<https://www.rfc-editor.org/info/rfc7471>>.
- [RFC7474] Bhatia, M., Hartman, S., Zhang, D., and A. Lindem, Ed., "Security Extension for OSPFv2 When Using Manual Key Management", RFC 7474, DOI 10.17487/RFC7474, April 2015, <<https://www.rfc-editor.org/info/rfc7474>>.

- [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>>.
- [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>>.

Authors' Addresses

Peter Psenak (editor)
Cisco Systems
Apollo Business Center
Mlynske nivy 43
Bratislava, 82109
Slovakia

Email: ppsenak@cisco.com

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

Email: shraddha@juniper.net

Clarence Filsfils
Cisco Systems, Inc.
Brussels
Belgium

Email: cfilsfil@cisco.com

Ketan Talaulikar
Arrcus, Inc
India

Email: ketant.ietf@gmail.com

Arkadiy Gulko
Edward Jones

Email: arkadiy.gulko@edwardjones.com

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: June 29, 2022

T. Li
Arista Networks
L. Ginsberg
P. Wells
Cisco Systems
December 26, 2021

IS-IS Extended Hierarchy
draft-ietf-lsr-isis-extended-hierarchy-05

Abstract

The IS-IS routing protocol was originally defined with a two level hierarchical structure. This was adequate for the networks at the time. As we continue to expand the scale of our networks, it is apparent that additional hierarchy would be a welcome degree of flexibility in network design.

This document defines IS-IS Levels 3 through 8.

Status of This Memo

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

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

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

This Internet-Draft will expire on June 29, 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Requirements Language	3
2. PDU changes	3
2.1. Circuit Type	4
2.2. PDU Type	4
3. Additional PDUs	4
3.1. Level n LAN IS to IS hello PDU (Ln-LAN-HELLO-PDU)	5
3.2. Level n Point-to-point IS to IS hello PDU (Ln-P2P-HELLO-PDU)	5
4. Level Specific Area Identifiers	5
4.1. IS-IS Area Hierarchy TLV	6
4.2. Adjacency Formation Rules	8
4.2.1. Level 3-8 Adjacency Formation Rules	8
4.2.2. Special Level-1 and Level-2 Adjacency Formation Rules	9
4.2.2.1. Actions on a Point-to-Point Circuit	9
4.2.2.2. Actions on a LAN Circuit	9
4.2.2.3. Reporting of Mismatched Area Hierarchies	9
5. New Flooding Scopes	10
6. MAC Addresses	12
7. Inheritance of TLVs	13
8. Behavior of Level n	13
9. Relationship between levels	13
10. Acknowledgements	13
11. IANA Considerations	13
11.1. PDU Type	13
11.2. New PDUs	13
11.3. New TLVs	14
11.4. New Flooding Scopes	14
11.5. New MAC Addresses	15
12. Security Considerations	16
13. Normative References	16
Appendix A. Preventing Cross Branching in the Hierarchy	16
Appendix B. Guidelines for Introducing a new level	18
Authors' Addresses	18

1. Introduction

The IS-IS routing protocol IS-IS [ISO10589] currently supports a two level hierarchy of abstraction. The fundamental unit of abstraction is the 'area', which is a (hopefully) connected set of systems

running IS-IS at the same level. Level 1, the lowest level, is abstracted by routers that participate in both Level 1 and Level 2.

Practical considerations, such as the size of an area's link state database, cause network designers to restrict the number of routers in any given area. Concurrently, the dominance of scale-out architectures based around small routers has created a situation where the scalability limits of the protocol are going to become critical in the foreseeable future.

The goal of this document is to enable additional hierarchy within IS-IS. Each additional level of hierarchy has a multiplicative effect on scale, so the addition of six levels should be a significant improvement. While all six levels may not be needed in the short term, it is apparent that the original designers of IS-IS reserved enough space for these levels, and defining six additional levels is only slightly harder than adding a single level, so it makes sense to expand the design for the future.

The modifications described herein are designed to be fully backward compatible and have no effect on existing networks. The modifications are also designed to have no effect whatsoever on networks that only use Level 1 and/or Level 2.

Section references in this document are references to sections of IS-IS [ISO10589].

Note that [ISO10589] uses a bit encoding convention where bit numbers are 1 based and Bit 1 is the Least Significant Bit (LSB) of the datatype. Traditionally IETF documents have used a bit encoding convention where bit numbers are 0 based and Bit 0 is the Most Significant Bit (MSB) of the datatype. This document uses [ISO10589] conventions throughout.

1.1. Requirements Language

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

2. PDU changes

In this section, we enumerate all of the redefinitions of protocol header fields necessary to add additional levels.

2.1. Circuit Type

In the fixed header of some IS-IS PDUs, a field is named 'Reserved/Circuit Type' (Section 9.5). The high order six bits are reserved, with the low order two bits indicating Level 1 (bit 1) and Level 2 (bit 2).

This field is renamed to be 'Circuit Type'. The bits are redefined as follows:

1. Level 1
2. Level 2
3. Level 3
4. Level 4
5. Level 5
6. Level 6
7. Level 7
8. Level 8

The value of zero (no bits set) is reserved. PDUs with a Circuit Type of zero SHALL be ignored.

The set bits of the Circuit Type MUST be contiguous. If bit n and bit m are set in the Circuit Type, then all bits in the interval [n:m] must be set.

2.2. PDU Type

The fixed header of IS-IS PDUs contains an octet with three reserved bits and the 'PDU Type' field. The three reserved bits are transmitted as zero and ignored on receipt. (Section 9.5)

To allow for additional PDU space, this entire octet is renamed the 'PDU Type' field.

3. Additional PDUs

3.1. Level n LAN IS to IS hello PDU (Ln-LAN-HELLO-PDU)

The 'Level n LAN IS to IS hello PDU' (Ln-LAN-HELLO-PDU) is identical in format to the 'Level 2 LAN IS to IS hello PDU' (Section 9.6), except that the PDU Types are defined as follows:

Level 3 (L3-LAN-HELLO-PDU): 33 (Suggested - to be assigned by IANA)

Level 4 (L4-LAN-HELLO-PDU): 34 (Suggested - to be assigned by IANA)

Level 5 (L5-LAN-HELLO-PDU): 35 (Suggested - to be assigned by IANA)

Level 6 (L6-LAN-HELLO-PDU): 36 (Suggested - to be assigned by IANA)

Level 7 (L7-LAN-HELLO-PDU): 37 (Suggested - to be assigned by IANA)

Level 8 (L8-LAN-HELLO-PDU): 38 (Suggested - to be assigned by IANA)

The Circuit Type field MUST be set to indicate all levels supported on that circuit - not just the level associated with the containing PDU type.

3.2. Level n Point-to-point IS to IS hello PDU (Ln-P2P-HELLO-PDU)

The 'Point-to-point IS to IS hello PDU' (Section 9.7) is used on Level 1 and Level 2 circuits. Legacy systems will not expect the circuit type field to indicate other levels, so a new PDU is used if the circuit supports other levels. The additional PDU is the 'Level n Point-to-point IS to IS hello PDU' (Ln-P2P-HELLO-PDU) and has PDU Type 39 (Suggested - to be assigned by IANA). The format of this PDU is identical to the existing Point-to-Point IS to IS hello PDU. Both PDUs may be used on the same circuit.

4. Level Specific Area Identifiers

[ISO10589] defines an Area Address to uniquely identify a Level-1 area. A given area may have multiple synonymous area addresses - which is useful in support of hitless merging or splitting of areas. Area address matching is part of the adjacency formation rules defined in Section 8 which determine whether a given adjacency supports Level-1, Level-2, or both. Area addresses are advertised in IIHs and LSPs using the Area Address TLV.

With the extensions defined in this document, there is a need to define an equivalent identifier for Levels 2-8. The Level Specific Area Identifier (LSAI) is a 16 bit value and is advertised using the new Area Hierarchy TLV defined in Section 4.1. There is no relationship between a Level-1 Area Address and an LSAI.

Just as with Area Addresses, multiple synonymous LSAIs may be assigned to a given level. This supports hitless merging or splitting of the level specific area. Although it is legal to do so, it is generally not useful to define more than two Area Identifiers for a given level.

A node MAY support any set of contiguous levels. Support for non-contiguous levels is undefined.

4.1. IS-IS Area Hierarchy TLV

The Area Hierarchy TLV specifies the set of LSAIs which comprise the branch of the network hierarchy to which the advertising node is connected. The TLV MUST include at least one LSAI for Levels 2-N, where N is ≥ 2 and N represents the highest level supported in the IS-IS domain. It is RECOMMENDED that $N == 8$ even when not all 8 levels are currently in use, but in cases where a network does not support higher levels a number less than 8 MAY be used.

Note that the levels advertised MAY include levels which are not supported by the advertising node.

The Area Hierarchy TLV has the following format:


```

      8 7 6 5 4 3 2 1
+---+---+---+---+---+---+
|   TLV Type   |
+---+---+---+---+---+---+
| TLV Length   |
+---+---+---+---+---+---+
| Supp-Levels   |
+---+---+---+---+---+---+

```

Followed by one or more Level Specific Area ID Sets:

```

      1                               0
      6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Level       | # of LSAIs   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+
|Level Specific Area Id(s) |
...
+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

TLV Type: ZZZ (1 octet)

TLV Length: Variable (1 octet)

Supp-Levels: A contiguous bitmask representing the set of levels supported by the advertising node (1 octet)

Bit #8 of this field is set if Level 8 is supported.
 Bit #7 of this field is set if Level 7 is supported.
 Bit #6 of this field is set if Level 6 is supported.
 Bit #5 of this field is set if Level 5 is supported.
 Bit #4 of this field is set if Level 4 is supported.
 Bit #3 of this field is set if Level 3 is supported.
 Bit #2 of this field is set if Level 2 is supported.
 Bit #1 of this field is set if Level 1 is supported

If the Supp-level bit mask is non-contiguous all advertised LSAIs are ignored.

Each Level Specific Area ID Set consists of:

Level: 2-8 (1 octet)

of LSAI: >=1 (1 octet)

LSAIs: The set of synonomous LSAIs associated with this level
 (2 * # of LSAIs octets)

The Area Hierarchy TLV MUST appear in all new IIH PDUs defined in Section 3. It MAY appear in P2P-HELLO-PDUs, L1-LAN-HELLO-PDUs, or L2-LAN-HELLO-PDUs.

The Area Hierarchy TLV MUST appear in LSP #0 of non-pseudo-node Level 3-8 Flooding Scoped LSPs defined in Section 5. It MAY appear in L1 or L2 LSP #0. It MUST NOT be present in any LSP with non-zero LSP number. If present in an LSP with non-zero LSP number it MUST be ignored on receipt.

Multiple Area Hierarchy TLVs MUST NOT be sent. In the event multiple Area Hierarchy TLVs are received, the first such TLV in the PDU is used. Subsequent TLVs in the same PDU MUST be ignored.

4.2. Adjacency Formation Rules

Adjacency formation rules for Levels 1 and 2 are defined in [ISO10589] and are not altered by these extensions except where noted below.

Adjacency Formation rules for Levels 3 and above are defined to insure that adjacency support for a given level is only enabled when there is a matching Area Identifier. Adjacency formation rules also are defined so as to prevent interconnection of neighbors which will connect to different areas at levels above any supported level.

The checks discussed below need to be performed on receipt of an IIH.

4.2.1. Level 3-8 Adjacency Formation Rules

The Area Hierarchy TLV MUST be present in a Level N Point-to-point IS to IS hello PDU or a Level N LAN IS to IS Hello PDU and the TLV content MUST adhere to the definition in Section 4.1. Beginning with the lowest level supported by the receiving node on this circuit and including all higher levels for which the receiver has an assigned LSAI regardless as to whether the higher levels are supported on this circuit, the set of LSAIs defined on the receiving node is compared against the set of LSAIs advertised in the received TLV. A matching LSAI MUST be found for each level.

If all of the checks pass then a new adjacency is formed or an existing adjacency is maintained.

NOTE: The absence of the advertisement of an LSAI for a given level is considered as a failure to find a matching LSAI.

On a Point-to-Point circuit, a single adjacency is formed which supports all of the levels supported by both nodes on this circuit.

On a LAN circuit, an adjacency is formed supporting only the level specified by the PDU type.

Note that (as previously specified) the set of levels advertised MUST be contiguous.

4.2.2. Special Level-1 and Level-2 Adjacency Formation Rules

The Area Hierarchy TLV MAY appear in a Point-to-point IS to IS hello PDU, Level 1 LAN IS to IS Hello PDU, or Level 2 LAN IS to IS Hello PDU (PDUs specified in [ISO10589]). In such a case, the neighbor may or may not support the Area Hierarchy TLV. The following sub-sections define modified adjacency formation rules for point-to-point and LAN circuits.

4.2.2.1. Actions on a Point-to-Point Circuit

If the Area Hierarchy TLV is present, then in addition to the checks specified in [ISO10589] the checks specified in Section 4.2.1 MUST be performed for all levels for which the receiver has an assigned LSAI beginning with Level 2. If those checks fail an adjacency MUST NOT be formed and any existing matching adjacency MUST transition to DOWN state.

4.2.2.2. Actions on a LAN Circuit

Adjacency formation MUST follow the rules defined in [ISO10589]. If the Area Hierarchy TLV is present in the Level 1 or Level 2 LAN IS to IS Hello PDU then the checks specified in Section 4.2.1 SHOULD be performed for all levels for which the receiver has an assigned LSAI beginning with Level 2. If those checks fail an error SHOULD be reported, but the level specific adjacency is still allowed. This prevents violation of the assumption of transitivity on the LAN in the presence of systems which do not support the extensions defined in this document.

4.2.2.3. Reporting of Mismatched Area Hierarchies

When forming adjacencies at Level-1 and/or Level-2, it is possible to have a mixture of legacy nodes (which do NOT support the extensions defined in this document) and new nodes which do support the extensions.

In Point-to-Point mode, legacy nodes will not advertise the new Area Hierarchy TLV and will not have an assigned LSAI for Level-2. It then becomes possible for new nodes with mismatched Area Hierarchies to form adjacencies with legacy nodes and form an L1 or L2 area where not all new nodes have a matching Area Hierarchy. This cannot be

detected when forming adjacencies if the new nodes are not directly connected - but it can be detected after the adjacencies have been formed by inspecting the set of Area Hierarchy TLVs in the level specific LSPs of all routers in the area.

Similarly in LAN mode, the transitivity requirement means that new nodes MUST form adjacencies with all nodes connected to the LAN even when the Area Hierarchy TLV mismatch check fails (see Section 4.2.2.2). This can occur both at Level-1 and Level-2.

New nodes MUST report these inconsistencies.

5. New Flooding Scopes

For levels 3-8, all link state information, PSNPs, and CSNPs are relayed in conformance with [RFC7356]. Additional flooding scopes are defined for each new level, for both circuit flooding scope and level flooding scope. Level flooding scopes are defined for both Standard and Extended TLV formats. The list of additional flooding scopes is:

Value	Description	FS LSP ID Format/ TLV Format
6	Level 3 Circuit Flooding Scope	Extended/Standard
7	Level 4 Circuit Flooding Scope	Extended/Standard
8	Level 5 Circuit Flooding Scope	Extended/Standard
9	Level 6 Circuit Flooding Scope	Extended/Standard
10	Level 7 Circuit Flooding Scope	Extended/Standard
11	Level 8 Circuit Flooding Scope	Extended/Standard
12	Level 3 Flooding Scope	Extended/Standard
13	Level 4 Flooding Scope	Extended/Standard
14	Level 5 Flooding Scope	Extended/Standard
15	Level 6 Flooding Scope	Extended/Standard
16	Level 7 Flooding Scope	Extended/Standard
17	Level 8 Flooding Scope	Extended/Standard
18	Level 3 Flooding Scope	Standard/Standard
19	Level 4 Flooding Scope	Standard/Standard
20	Level 5 Flooding Scope	Standard/Standard
21	Level 6 Flooding Scope	Standard/Standard
22	Level 7 Flooding Scope	Standard/Standard
23	Level 8 Flooding Scope	Standard/Standard
70	Level 3 Circuit Flooding Scope	Extended/Extended
71	Level 4 Circuit Flooding Scope	Extended/Extended
72	Level 5 Circuit Flooding Scope	Extended/Extended
73	Level 6 Circuit Flooding Scope	Extended/Extended
74	Level 7 Circuit Flooding Scope	Extended/Extended
75	Level 8 Circuit Flooding Scope	Extended/Extended
76	Level 3 Flooding Scope	Extended/Extended
77	Level 4 Flooding Scope	Extended/Extended
78	Level 5 Flooding Scope	Extended/Extended
79	Level 6 Flooding Scope	Extended/Extended
80	Level 7 Flooding Scope	Extended/Extended
81	Level 8 Flooding Scope	Extended/Extended

The final octet of the header of a Flooding Scoped LSP as defined in [RFC7356] contains Reserved/LSPDBOL/IS Type information. This field is redefined for the new flooding scopes defined in this document as follows:

Reserved/ATT/LSPDBOL

Bits 8-5 Reserved

Transmitted as 0 and ignored on receipt

Bit 4 ATT

If set to 1 indicates that the sending IS is attached to
routers in other Level N areas via Level N+1

Bit 3 LSDBOL

As defined in RFC7356

Bits 2-1

Transmitted as 0 and ignored on receipt.

Note that the levels supported (analogous to the IS-type information in L1 and L2 LSPs) can be obtained from the Area Hierarchy TLV advertised in the associated LSP #0.

Note that the definition of the ATT bit specified above also applies to L2 LSPs. Previously this bit would have no meaning as [ISO10589] does not define support for Level 3.

6. MAC Addresses

On a broadcast network, PDUs are currently sent to the AllL1Iss or AllL2Iss MAC addresses. We will need additional MAC addresses for Levels 3-8.

AllL3ISs: MAC3

AllL4ISs: MAC4

AllL5ISs: MAC5

AllL6ISs: MAC6

AllL7ISs: MAC7

AllL8ISs: MAC8

When operating in Point-to-Point mode on a broadcast network [RFC5309], a Level N Point-to-Point Hello PDU will be sent. Any of the above MAC addresses could be used in this case, but it is recommended to use the AllL3ISs MAC address.

7. Inheritance of TLVs

All existing Level 2 TLVs may be used in the corresponding Level 3 through Level 8 PDUs. When used in a Level 3 through Level 8 PDU, the semantics of these TLVs will be applied to the Level of the containing PDU. If the original semantics of the PDU was carrying a reference to Level 1 in a Level 2 TLV, then the semantics of the TLV at level N will be a reference to level N-1. The intent is to retain the original semantics of the TLV at the higher level.

8. Behavior of Level n

The behavior of Level n is analogous to the behavior of Level 2.

9. Relationship between levels

The relationship between Level n and Level n-1 is analogous to the relationship between Level 2 and Level 1.

An area at Level n has at most one parent at Level n+1.

10. Acknowledgements

The authors would like to thank Dinesh Dutt for inspiring this document and Huaimo Chen for his comments. The authors would also like to thank Tony Pryzienda for his careful review and excellent suggestions.

11. IANA Considerations

This document makes many requests to IANA, as follows:

11.1. PDU Type

The existing IS-IS PDU registry currently supports values 0-31. This should be expanded to support the values 0-255. The existing value assignments should be retained. Value 255 should be reserved.

11.2. New PDUs

IANA is requested to allocate values from the IS-IS PDU registry for the following:

L3-LAN-HELLO-PDU: 33 (Suggested - to be assigned by IANA)

L4-LAN-HELLO-PDU: 34 (Suggested - to be assigned by IANA)

L5-LAN-HELLO-PDU: 35 (Suggested - to be assigned by IANA)

L6-LAN-HELLO-PDU: 36 (Suggested - to be assigned by IANA)

L7-LAN-HELLO-PDU: 37 (Suggested - to be assigned by IANA)

L8-LAN-HELLO-PDU: 38 (Suggested - to be assigned by IANA)

Ln-P2P-HELLO-PDU: 39 (Suggested - to be assigned by IANA)

11.3. New TLVs

IANA is requested to allocate values from the IS-IS TLV registry for the following:

Area Hierarchy: ZZZ

11.4. New Flooding Scopes

IANA is requested to allocate the following values from the IS-IS Flooding Scope Identifier Registry.

Value	Description	FS LSP ID Format/ TLV Format	IIH Announce Lx-P2P Lx-LAN
6	Level 3 Circuit Flooding Scope	Extended/Standard	Y Y
7	Level 4 Circuit Flooding Scope	Extended/Standard	Y Y
8	Level 5 Circuit Flooding Scope	Extended/Standard	Y Y
9	Level 6 Circuit Flooding Scope	Extended/Standard	Y Y
10	Level 7 Circuit Flooding Scope	Extended/Standard	Y Y
11	Level 8 Circuit Flooding Scope	Extended/Standard	Y Y
12	Level 3 Flooding Scope	Extended/Standard	Y Y
13	Level 4 Flooding Scope	Extended/Standard	Y Y
14	Level 5 Flooding Scope	Extended/Standard	Y Y
15	Level 6 Flooding Scope	Extended/Standard	Y Y
16	Level 7 Flooding Scope	Extended/Standard	Y Y
17	Level 8 Flooding Scope	Extended/Standard	Y Y
18	Level 3 Flooding Scope	Standard/Standard	Y Y
19	Level 4 Flooding Scope	Standard/Standard	Y Y
20	Level 5 Flooding Scope	Standard/Standard	Y Y
21	Level 6 Flooding Scope	Standard/Standard	Y Y
22	Level 7 Flooding Scope	Standard/Standard	Y Y
23	Level 8 Flooding Scope	Standard/Standard	Y Y
70	Level 3 Circuit Flooding Scope	Extended/Extended	Y Y
71	Level 4 Circuit Flooding Scope	Extended/Extended	Y Y
72	Level 5 Circuit Flooding Scope	Extended/Extended	Y Y
73	Level 6 Circuit Flooding Scope	Extended/Extended	Y Y
74	Level 7 Circuit Flooding Scope	Extended/Extended	Y Y
75	Level 8 Circuit Flooding Scope	Extended/Extended	Y Y
76	Level 3 Flooding Scope	Extended/Extended	Y Y
77	Level 4 Flooding Scope	Extended/Extended	Y Y
78	Level 5 Flooding Scope	Extended/Extended	Y Y
79	Level 6 Flooding Scope	Extended/Extended	Y Y
80	Level 7 Flooding Scope	Extended/Extended	Y Y
81	Level 8 Flooding Scope	Extended/Extended	Y Y

11.5. New MAC Addresses

IANA is requested to allocate values from the IANA Multicast 48-bit MAC Addresses block for the following:

AllL3Iss: MAC3

AllL4Iss: MAC4

AllL5Iss: MAC5

AllL6Iss: MAC6

AllL7Iss: MAC7

AllL8Iss: MAC8

12. Security Considerations

This document introduces no new security issues. Security of routing within a domain is already addressed as part of the routing protocols themselves. This document proposes no changes to those security architectures.

13. Normative References

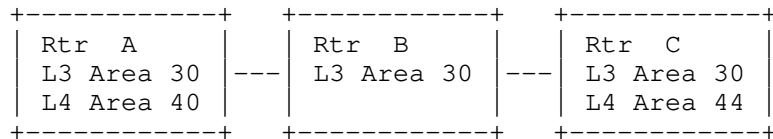
- [ISO10589] International Organization for Standardization, "Intermediate System to Intermediate System Intra-Domain Routing Exchange Protocol for use in Conjunction with the Protocol for Providing the Connectionless-mode Network Service (ISO 8473)", ISO/IEC 10589:2002, November 2002.
- [RFC2119] Bradner, 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>>.
- [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>>.
- [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>>.
- [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>>.

Appendix A. Preventing Cross Branching in the Hierarchy

The use of additional levels requires careful interconnection of routers which support multiple levels. Consistent association of LSAs is required not only for validating the connections between routers in a level specific area but also for all levels above a given level to which any of the routers may be connected (directly or indirectly). Failure to do so can result in interconnecting different branches of a tree leading to interarea loops. This leads to the requirement that all routers advertise an LSAI for all levels

regardless of whether a given router is configured to participate in a given level or not.

At first glance it may seem that it would be sufficient for each router to advertise LSAIs only for the levels that the router is configured to support. However, the following simple example illustrates why this is problematic.



Since Router B does not support Level 4, it chose not to advertise any Area for Level 4. This means that neither Router A nor Router C can tell by inspecting hellos that not all routers in Level 3 area 30 have been configured to support the same Level 4 area. It is possible for Rtr A and Rtr C to discover the LSAIs advertised by all routers by inspecting the Level 3 LSPs - however this requires that Level 3 adjacencies be formed and maintained even when routing cannot be safely performed via all adjacencies in a given area. It then needs to be decided how routing over existing adjacencies should be limited. A number of possibilities exist:

Treat the area as if it were two partitions. In the example Router A would be in one partition and Router C would be in another partition. But Router B could belong to either partition.

Select a winning Level 4 Area among the set of Level 4 areas advertised in L3 LSPs and only allow leaking of routes to/from that level

But either of these options introduce the possibility that a previously fully connected hierarchy becomes partially disconnected as a result of a single configuration change on a single router and/or the bringup of a new router.

The choice made was then to require all routers supporting the extensions in this document to advertise an LSAI for all levels regardless of what specific levels an individual router is configured to support. This guarantees that any inconsistency between the intended connectivity of a router at all levels - direct and indirect - can be detected during exchange of hellos and therefore adjacency bringup can always be blocked when necessary.

Appendix B. Guidelines for Introducing a new level

It is desirable to be able to introduce support for a new level without disruption. This section discusses ways to do this.

Initial deployment may require only the support of one additional level (Level 3). However, in the future increased network scale may make introduction of an additional level (Level 4) desirable. It is suggested that all routers be configured to advertise a single candidate LSAI for Level 4 - for the purposes of the example let's use LSAI 44. When ready to deploy Level 4, it is then only necessary to enable Level 4 on those routers who will be participating in the additional level.

However, perhaps at the time of deploying Level 3 the administrator has no idea what LSAI will be used for Level 4 in the future. In such a case a "dummy" LSAI should be configured for Level 4 on all routers - let's use "0" in this example. In this case, what needs to be done when ready to enable Level 4 is to go to every router (regardless of whether it will actively participate in the new level) and configure the intended LSAI for Level 4. If LSAI 45 is the intended Level 4 area, then LSAI 45 is configured on each router. Each router is then advertising two LSAIs for Level 4: (0, 45). Once this is completed, go to every router and remove the "dummy" Level 4 LSAI (0) and the network is now ready to have this Level 4 area enabled.

In the event that support for a new level needs to be introduced and no LSAI was ever advertised for that level, the introduction of LSAI for the new level will cause temporary adjacency flaps as the advertisement of the LSAI for the new level is introduced. To avoid this, implementations would need to introduce support for temporary disablement of the LSAI check for the new level until the configuration of the new LSAI is complete on all nodes. Support for this transition mode is outside the scope of this document. The need for a transition mode can be avoided if an LSAI is configured for levels 2-8 from day one.

Authors' Addresses

Tony Li
Arista Networks
5453 Great America Parkway
Santa Clara, California 95054
United States of America

Email: tony.li@tony.li

Les Ginsberg
Cisco Systems
United States of America

Email: ginsberg@cisco.com

Paul Wells
Cisco Systems
United States of America

Email: pauwells@cisco.com

Networking Working Group
Internet-Draft
Updates: 7370 (if approved)
Intended status: Standards Track
Expires: April 23, 2022

P. Psenak, Ed.
C. Filsfils
Cisco Systems
A. Bashandy
Individual
B. Decraene
Orange
Z. Hu
Huawei Technologies
October 20, 2021

IS-IS Extensions to Support Segment Routing over IPv6 Dataplane
draft-ietf-lsr-isis-srv6-extensions-18

Abstract

The Segment Routing (SR) architecture allows flexible definition of the end-to-end path by encoding it as a sequence of topological elements called "segments". It can be implemented over the MPLS or the IPv6 data plane. This document describes the IS-IS extensions required to support Segment Routing over the IPv6 data plane.

This document updates RFC 7370 by modifying an existing registry.

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

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

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

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

This Internet-Draft will expire on April 23, 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
2. SRv6 Capabilities sub-TLV	4
3. Advertising Supported Algorithms	5
4. Advertising Maximum SRv6 SID Depths	5
4.1. Maximum Segments Left MSD Type	5
4.2. Maximum End Pop MSD Type	5
4.3. Maximum H.Encaps MSD Type	5
4.4. Maximum End D MSD Type	6
5. SRv6 SIDs and Reachability	6
6. Advertising Anycast Property	8
7. Advertising Locators and End SIDs	9
7.1. SRv6 Locator TLV Format	9
7.2. SRv6 End SID sub-TLV	11
8. Advertising SRv6 Adjacency SIDs	12
8.1. SRv6 End.X SID sub-TLV	13
8.2. SRv6 LAN End.X SID sub-TLV	14
9. SRv6 SID Structure Sub-Sub-TLV	16
10. Advertising Endpoint Behaviors	17
11. IANA Considerations	18
11.1. SRv6 Locator TLV	18
11.1.1. SRv6 End SID sub-TLV	18
11.1.2. Revised sub-TLV table	19
11.2. SRv6 Capabilities sub-TLV	19
11.3. Sub-Sub-TLVs of the SRv6 Capability sub-TLV	20
11.4. SRv6 End.X SID and SRv6 LAN End.X SID sub-TLVs	20
11.5. MSD Types	20
11.6. Sub-Sub-TLVs for SID Sub-TLVs	21
11.7. Prefix Attribute Flags Sub-TLV	21
11.8. ISIS SRv6 Capabilities sub-TLV Flags Registry	21

11.9. ISIS SRv6 Locator TLV Flags Registry	22
11.10. ISIS SRv6 End SID sub-TLV Flags Registry	22
11.11. ISIS SRv6 End.X SID and LAN End.X SID sub-TLVs Flags Registry	23
12. Security Considerations	23
13. Contributors	24
14. Acknowledgments	25
15. References	26
15.1. Normative References	26
15.2. Informative References	28
Authors' Addresses	28

1. Introduction

With Segment Routing (SR) [RFC8402], a node steers a packet through an ordered list of instructions, called segments.

Segments are identified through Segment Identifiers (SIDs).

Segment Routing can be directly instantiated on the IPv6 data plane through the use of the Segment Routing Header defined in [RFC8754]. SRv6 refers to this SR instantiation on the IPv6 dataplane.

The network programming paradigm [RFC8986] is central to SRv6. It describes how any behavior can be bound to a SID and how any network program can be expressed as a combination of SIDs.

This document specifies IS-IS extensions that allow the IS-IS protocol to encode some of these SIDs and their behaviors.

Familiarity with the network programming paradigm [RFC8986] is necessary to understand the extensions specified in this document.

The new SRv6 Locator top level TLV announces SRv6 locators - a form of summary address for the set of topology/algorithm-specific SIDs instantiated at the node.

The SRv6 Capabilities sub-TLV announces the ability to support SRv6.

Several new sub-TLVs are defined to advertise various SRv6 Maximum SID Depths.

The SRv6 End SID sub-TLV, the SRv6 End.X SID sub-TLV, and the SRv6 LAN End.X SID sub-TLV are used to advertise which SIDs are instantiated at a node and what Endpoint behavior is bound to each instantiated SID.

This document updates [RFC7370] by modifying an existing registry (Section 11.1.2).

2. SRv6 Capabilities sub-TLV

A node indicates that it supports the SR Segment Endpoint Node functionality as specified in [RFC8754] by advertising a new SRv6 Capabilities sub-TLV of the router capabilities TLV [RFC7981].

The SRv6 Capabilities sub-TLV may contain optional sub-sub-TLVs. No sub-sub-TLVs are currently defined.

The SRv6 Capabilities 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																			
optional sub-sub-TLVs...																																							

Type: 25. Single octet as defined in section 9 of [ISO10589].

Length: Single octet as defined in section 9 of [ISO10589]. The length value is 2 + length of sub-sub-TLVs.

Flags: 2 octets The following flags are defined:

0										1									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5				
0										Reserved									

where:

O-flag: If set, the router supports use of the O-bit in the Segment Routing Header (SRH) as defined in [I-D.ietf-6man-spring-srv6-oam].

The remaining bits, including bit 0, are reserved for future use. They MUST be set to zero on transmission and MUST be ignored on receipt.

3. Advertising Supported Algorithms

An SRv6 capable router indicates supported algorithm(s) by advertising the Segment Routing Algorithm sub-TLV as defined in [RFC8667].

4. Advertising Maximum SRv6 SID Depths

[RFC8491] defines the means to advertise node/link specific values for Maximum SID Depths (MSD) of various types. Node MSDs are advertised in a sub-TLV of the Router Capabilities TLV [RFC7981]. Link MSDs are advertised in a sub-TLV of TLVs 22, 23, 25, 141, 222, and 223.

This document defines the relevant SRv6 MSDs and requests MSD type assignments in the MSD Types registry created by [RFC8491].

4.1. Maximum Segments Left MSD Type

The Maximum Segments Left MSD Type signals the maximum value of the "Segments Left" field [RFC8754] in the SRH of a received packet before applying the Endpoint behavior associated with a SID.

SRH Max Segments Left Type: 41

If no value is advertised, the supported value is 0.

4.2. Maximum End Pop MSD Type

The Maximum End Pop MSD Type signals the maximum number of SIDs in the SRH to which the router can apply "Penultimate Segment Pop of the SRH" or "Ultimate Segment Pop of the SRH" behavior, as defined in [RFC8986] flavors.

SRH Max End Pop Type: 42

If the advertised value is zero or no value is advertised, then the router cannot apply PSP or USP flavors.

4.3. Maximum H.Encaps MSD Type

The Maximum H.Encaps MSD Type signals the maximum number of SIDs that can be added to the Segment List of an SRH as part of the "H.Encaps" behavior as defined in [RFC8986].

SRH Max H.encaps Type: 44

If the advertised value is zero or no value is advertised, then the headend can apply an SR Policy that only contains one segment, without inserting any SRH header.

A non-zero SRH Max H.encaps MSD indicates that the headend can insert an SRH up to the advertised number of SIDs.

4.4. Maximum End D MSD Type

The Maximum End D MSD Type specifies the maximum number of SIDs present in an SRH when performing decapsulation. As specified in [RFC8986] the permitted SID types include, but are not limited to End.DX6, End.DT4, End.DT46, End with USD, End.X with USD.

SRH Max End D Type: 45

If the advertised value is zero or no value is advertised then the router cannot apply any behavior that results in decapsulation and forwarding of the inner packet if the outer IPv6 header contains an SRH.

5. SRv6 SIDs and Reachability

As discussed in [RFC8986], an SRv6 Segment Identifier (SID) is 128 bits and consists of Locator, Function and Argument parts.

A node is provisioned with topology/algorithm specific locators for each of the topology/algorithm pairs supported by that node. Each locator is a covering prefix for all SIDs provisioned on that node which have the matching topology/algorithm.

Locators MUST be advertised in the SRv6 Locator TLV (see Section 7.1). Forwarding entries for the locators advertised in the SRv6 Locator TLV MUST be installed in the forwarding plane of receiving SRv6 capable routers when the associated topology/algorithm is supported by the receiving node. The processing of the prefix advertised in the SRv6 Locator TLV, the calculation of its reachability and the installation in the forwarding plane follows the process defined for the Prefix Reachability TLV 236 [RFC5308], or TLV 237 [RFC5120].

Locators associated with algorithm 0 and 1 (for all supported topologies) SHOULD be advertised in a Prefix Reachability TLV (236 or 237) so that legacy routers (i.e., routers which do not support SRv6) will install a forwarding entry for algorithm 0 and 1 SRv6 traffic.

In cases where the same prefix, with the same prefix-length, Multi Topology ID (MT ID), and algorithm is received in both a Prefix Reachability TLV and an SRv6 Locator TLV, the Prefix Reachability advertisement MUST be preferred when installing entries in the forwarding plane. This is to prevent inconsistent forwarding entries between SRv6 capable and SRv6 incapable routers. Such preference of Prefix Reachability advertisement does not have any impact on the rest of the data advertised in the SRv6 Locator TLV.

Locators associated with Flexible Algorithms (see Section 4 of [I-D.ietf-lsr-flex-algo]) SHOULD NOT be advertised in Prefix Reachability TLVs (236 or 237). Advertising the Flexible Algorithm locator in regular Prefix Reachability TLV (236 or 237) would make the forwarding for it to follow algo 0 path.

SRv6 SIDs are advertised as sub-TLVs in the SRv6 Locator TLV except for SRv6 SIDs which are associated with a specific Neighbor/Link and are therefore advertised as sub-TLVs in TLVs 22, 23, 25, 141, 222, and 223.

SRv6 SIDs received from other nodes are not directly routable and MUST NOT be installed in the forwarding plane. Reachability to SRv6 SIDs depends upon the existence of a covering locator.

Adherence to the rules defined in this section will assure that SRv6 SIDs associated with a supported topology/algorithm pair will be forwarded correctly, while SRv6 SIDs associated with an unsupported topology/algorithm pair will be dropped. NOTE: The drop behavior depends on the absence of a default/summary route covering a given locator.

In order for forwarding to work correctly, the locator associated with SRv6 SID advertisements must be the longest match prefix installed in the forwarding plane for those SIDs. In order to ensure correct forwarding, network operators should take steps to make sure that this requirement is not compromised. For example, the following situations should be avoided:

- o Another locator associated with a different topology/algorithm is the longest match
- o Another prefix advertisement (i.e., from TLV 236 or 237) is the longest match

6. Advertising Anycast Property

Both prefixes and SRv6 Locators may be configured as anycast and as such the same value can be advertised by multiple routers. It is useful for other routers to know that the advertisement is for an anycast identifier.

A new flag in Prefix Attribute Flags Sub-TLV [RFC7794] is defined to advertise the anycast property:

Bit #: 4
Name: Anycast Flag (A-flag)

When the prefix/SRv6 locator is configured as anycast, the A-flag SHOULD be set. Otherwise, this flag MUST be clear.

The A-flag MUST be preserved when the advertisement is leaked between levels.

The A-flag and the N-flag MUST NOT both be set. If both N-flag and A-flag are set in the prefix/SRv6 Locator advertisement, the receiving routers MUST ignore the N-flag.

The same prefix/SRv6 Locator can be advertised by multiple routers. If at least one of them sets the A-Flag in its advertisement, the prefix/SRv6 Locator SHOULD be considered as anycast.

A prefix/SRv6 Locator that is advertised by a single node and without an A-Flag is considered node specific.

All the nodes advertising the same anycast locator MUST instantiate the exact same set of SIDs under that anycast locator. Failure to do so may result in traffic being black-holed or mis-routed.

The Prefix Attribute Flags Sub-TLV can be carried in the SRv6 Locator TLV as well as the Prefix Reachability TLVs. When a router originates both the Prefix Reachability TLV and the SRv6 Locator TLV for a given prefix, and the router is originating the Prefix Attribute Flags Sub-TLV in one of the TLVs, the router SHOULD advertise the same flags in the Prefix Attribute Flags Sub-TLV in both TLVs. However, unlike TLVs 236 [RFC5308] and 237 [RFC5120] the X-flag in the Prefix Attributes Flags sub-TLV is valid when sent in the SRv6 Locator TLV. The state of the X-flag in the Prefix Attributes Flags sub-TLV when included in the Locator TLV MUST match the setting of the embedded "X-bit" in any advertisement for the same prefix in TLVs 236 [RFC5308] and 237 [RFC5120]. In case of any inconsistency between the Prefix Attribute Flags advertised in the

Locator TLV and in the Prefix Reachability TLV, the ones advertised in Prefix Reachability TLV MUST be preferred.

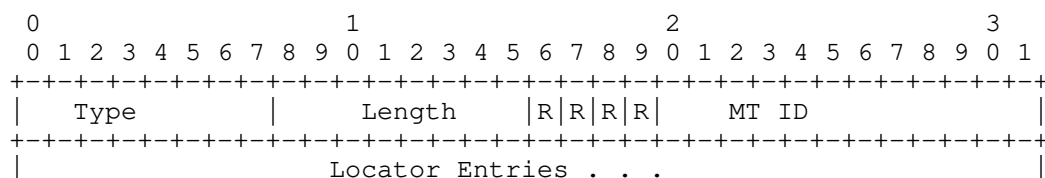
7. Advertising Locators and End SIDs

The SRv6 Locator TLV is introduced to advertise SRv6 Locators and End SIDs associated with each locator.

This new TLV shares the sub-TLV space defined for TLVs 135, 235, 236 and 237.

7.1. SRv6 Locator TLV Format

The SRv6 Locator TLV has the following format:



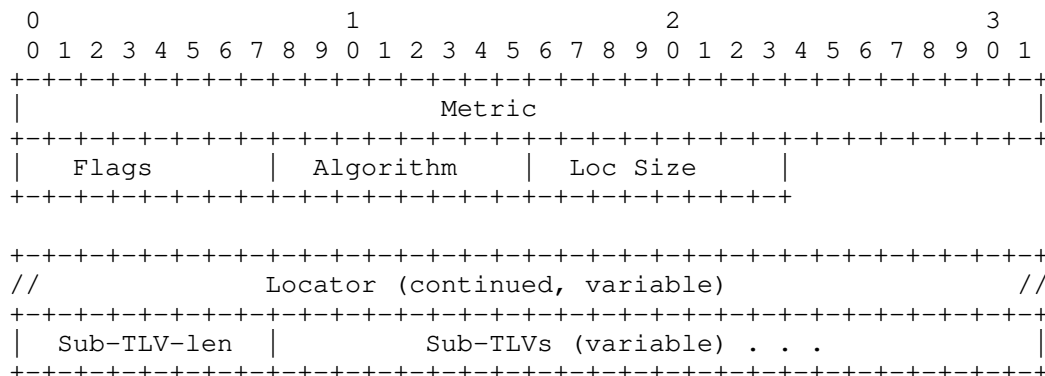
Type: 27. Single octet as defined in section 9 of [ISO10589].

Length: Single octet as defined in section 9 of [ISO10589]. The length value is variable.

R bits: reserved for future use. They MUST be set to zero on transmission and MUST be ignored on receipt.

MT ID: Multitopology Identifier as defined in [RFC5120]. Note that the value 0 is legal.

Followed by one or more locator entries of the form:



Metric: 4 octets. As described in Section 4 of [RFC5305].

Flags: 1 octet. The following flags are defined:

```

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

D-flag: Same as described in section 4.1. of [RFC5305].

The remaining bits are reserved for future use. They MUST be set to zero on transmission and MUST be ignored on receipt.

Algorithm: 1 octet. As defined in IGP Algorithm Types registry [RFC8665].

Loc-Size: 1 octet. Number of bits in the SRv6 Locator field. MUST be from the range (1 - 128). The TLV MUST be ignored if the Loc-Size is outside this range.

Locator: 1-16 octets. This field encodes the advertised SRv6 Locator. The Locator is encoded in the minimal number of octets for the given number of bits. Trailing bits MUST be set to zero and ignored when received.

Sub-TLV-length: 1 octet. Number of octets used by sub-TLVs.

Optional sub-TLVs: Supported sub-TLVs are specified in Section 11.1.2. Any Sub-TLV that is not allowed in the SRv6 Locator TLV MUST be ignored.

Prefix Attribute Flags Sub-TLV [RFC7794] SHOULD be included in the Locator TLV.

Prefix Attribute Flags Sub-TLV MUST be included in the the Locator TLV when it is leaked upwards in the hierarchy or originated as a result of the redistribution from another protocol or another ISIS instance. If the Prefix Attribute Flags Sub-TLV is not included in these cases, receivers will be unable to determine the correct source of the advertisement. The receivers will be unable to detect the violation.

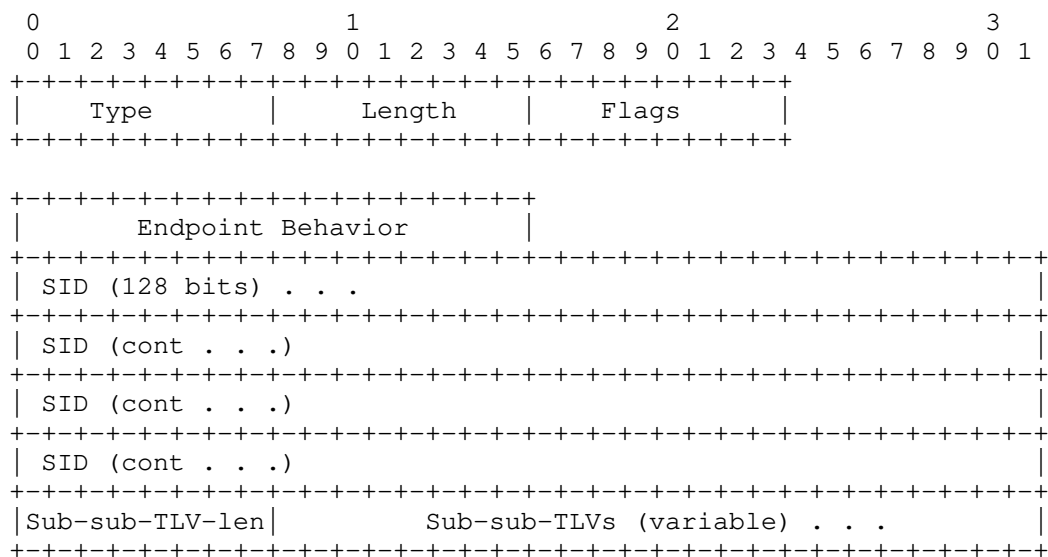
7.2. SRv6 End SID sub-TLV

The SRv6 End SID sub-TLV is introduced to advertise SRv6 Segment Identifiers (SID) with Endpoint behaviors which do not require a particular neighbor in order to be correctly applied. SRv6 SIDs associated with a neighbor are advertised using the sub-TLVs defined in Section 8.

Supported behavior values, together with parent TLVs in which they are advertised, are specified in Section 10 of this document. Please note that not all behaviors defined in [RFC8986] are defined in this document, e.g. END.T is not.

This new sub-TLV is advertised in the SRv6 Locator TLV defined in the previous section. SRv6 End SIDs inherit the topology/algorithm from the parent locator.

The SRv6 End SID sub-TLV has the following format:



Type: 5. Single octet as defined in section 9 of [ISO10589].

Length: Single octet as defined in section 9 of [ISO10589]. The length value is variable.

Flags: 1 octet. No flags are currently defined. All bits are reserved for future use. They MUST be set to zero on transmission and MUST be ignored on receipt.

Endpoint Behavior: 2 octets, as defined in [RFC8986]. Supported behavior values for this sub-TLV are defined in Section 10 of this document. Unsupported or unrecognized behavior values are ignored by the receiver.

SID: 16 octets. This field encodes the advertised SRv6 SID.

Sub-sub-TLV-length: 1 octet. Number of octets used by sub-sub-TLVs.

Optional Sub-sub-TLVs: Supported Sub-sub-TLVs are specified in Section 11.6. Any Sub-sub-TLV that is not allowed in SRv6 End SID sub-TLV MUST be ignored.

The SRv6 End SID MUST be allocated from its associated locator. SRv6 End SIDs that are not allocated from the associated locator MUST be ignored.

Multiple SRv6 End SIDs MAY be associated with the same locator. In cases where the number of SRv6 End SID sub-TLVs exceeds the capacity of a single TLV, multiple Locator TLVs for the same locator MAY be advertised. For a given MTID/Locator the algorithm MUST be the same in all TLVs. If this restriction is not met all TLVs for that MTID/Locator MUST be ignored.

8. Advertising SRv6 Adjacency SIDs

Certain SRv6 Endpoint behaviors [RFC8986] are associated with a particular adjacency.

This document defines two new sub-TLVs of TLV 22, 23, 25, 141, 222, and 223 - namely "SRv6 End.X SID sub-TLVs" and "SRv6 LAN End.X SID sub-TLVs".

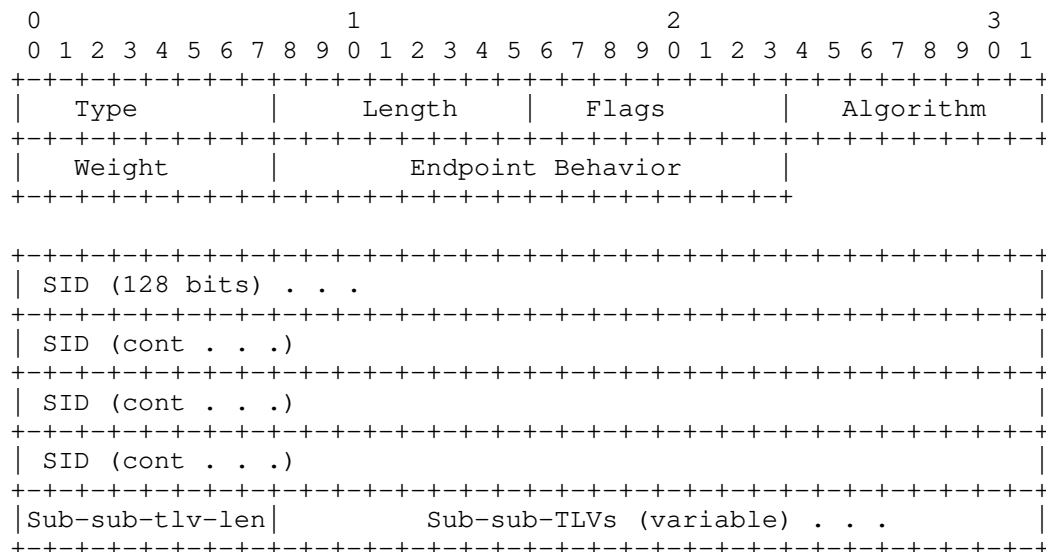
IS-IS Neighbor advertisements are topology specific - but not algorithm specific. SIDs advertised in SRv6 End.X SID and SRv6 LAN End.X SID sub-TLVs therefore inherit the topology from the associated neighbor advertisement, but the algorithm is specified in the individual SID.

All SIDs advertised in SRv6 End.X SID and SRv6 LAN End.X SID sub-TLVs MUST be a subnet of a Locator with matching topology and algorithm which is advertised by the same node in an SRv6 Locator TLV. SIDs that do not meet this requirement MUST be ignored. This ensures that the node advertising these SIDs is also advertising its corresponding Locator with the algorithm that will be used for computing paths destined to the SID.

8.1. SRv6 End.X SID sub-TLV

This sub-TLV is used to advertise an SRv6 SID associated with a point to point adjacency. Multiple SRv6 End.X SID sub-TLVs MAY be associated with the same adjacency.

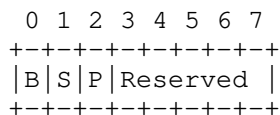
The SRv6 End.X SID sub-TLV has the following format:



Type: 43. Single octet as defined in section 9 of [ISO10589].

Length: Single octet as defined in section 9 of [ISO10589]. The length value is variable.

Flags: 1 octet.



where:

B-Flag: Backup flag. If set, the SID is eligible for protection, e.g., using IP Fast Re-route (IPFRR) [RFC5286], as described in [RFC8355].

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

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

Algorithm: 1 octet. As defined in IGP Algorithm Types registry [RFC8665].

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

Endpoint Behavior: 2 octets. As defined in [RFC8986]. Supported behavior values for this sub-TLV are defined in Section 10 of this document. Unsupported or unrecognized behavior values are ignored by the receiver.

SID: 16 octets. This field encodes the advertised SRv6 SID.

Sub-sub-TLV-length: 1 octet. Number of octets used by sub-sub-TLVs.

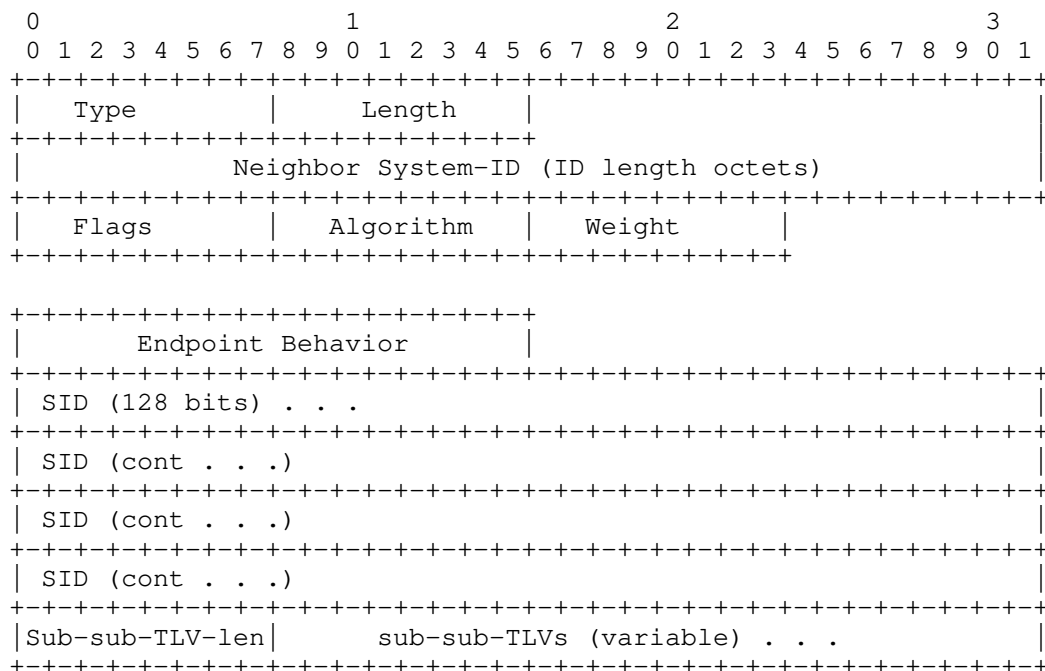
Optional Sub-sub-TLVs: Supported Sub-sub-TLVs are specified in Section 11.6. Any Sub-sub-TLV that is not allowed in SRv6 End.X SID sub-TLV MUST be ignored.

Note that multiple TLVs for the same neighbor may be required in order to advertise all the SRv6 SIDs associated with that neighbor.

8.2. SRv6 LAN End.X SID sub-TLV

This sub-TLV is used to advertise an SRv6 SID associated with a LAN adjacency. Since the parent TLV is advertising an adjacency to the Designated Intermediate System (DIS) for the LAN, it is necessary to include the System ID of the physical neighbor on the LAN with which the SRv6 SID is associated. Given that many neighbors may exist on a given LAN, multiple SRv6 LAN END.X SID sub-TLVs may be associated with the same LAN. Note that multiple TLVs for the same DIS neighbor may be required in order to advertise all the SRv6 SIDs associated with that neighbor.

The SRv6 LAN End.X SID sub-TLV has the following format:

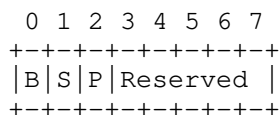


Type: 44. Single octet as defined in section 9 of [ISO10589].

Length: Single octet as defined in section 9 of [ISO10589]. The length value is variable.

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

Flags: 1 octet.



where B,S, and P flags are as described in Section 8.1.
Reserved bits MUST be zero when originated and MUST be ignored when received.

Algorithm: 1 octet. As defined in IGP Algorithm Types registry [RFC8665].

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

Endpoint Behavior: 2 octets. As defined in [RFC8986]. Supported behavior values for this sub-TLV are defined in Section 10 of this document. Unsupported or unrecognized behavior values are ignored by the receiver.

SID: 16 octets. This field encodes the advertised SRv6 SID.

Sub-sub-TLV-length: 1 octet. Number of octets used by sub-sub-TLVs.

Optional Sub-sub-TLVs: Supported Sub-sub-TLVs are specified in Section 11.6. Any Sub-sub-TLV that is not allowed in SRv6 LAN End.X SID sub-TLV MUST be ignored.

Note that multiple TLVs for the same neighbor, on the same LAN, may be required in order to advertise all the SRv6 SIDs associated with that neighbor.

9. SRv6 SID Structure Sub-Sub-TLV

SRv6 SID Structure Sub-Sub-TLV is an optional Sub-Sub-TLV of:

SRv6 End SID Sub-TLV (Section 7.2)

SRv6 End.X SID Sub-TLV (Section 8.1)

SRv6 LAN End.X SID Sub-TLV (Section 8.2)

SRv6 SID Structure Sub-Sub-TLV is used to advertise the structure of the SRv6 SID as defined in [RFC8986]. It 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																							
LB Length								LN Length								Fun. Length								Arg. Length							

where:

Type: 1. Single octet as defined in section 9 of [ISO10589].

Length: Single octet as defined in section 9 of [ISO10589]. The length value is 4 octets.

LB Length: 1 octet. SRv6 SID Locator Block length in bits.

LN Length: 1 octet. SRv6 SID Locator Node length in bits.

Fun. Length: 1 octet. SRv6 SID Function length in bits.

Arg. Length: 1 octet. SRv6 SID Arguments length in bits.

ISIS SRv6 SID Structure Sub-Sub-TLV MUST NOT appear more than once in its parent Sub-TLV. If it appears more than once in its parent Sub-TLV, the parent Sub-TLV MUST be ignored by the receiver.

The sum of all four sizes advertised in ISIS SRv6 SID Structure Sub-Sub-TLV MUST be less than or equal to 128 bits. If the sum of all four sizes advertised in the ISIS SRv6 SID Structure Sub-Sub-TLV is larger than 128 bits, the parent Sub-TLV MUST be ignored by the receiver.

The SRv6 SID Structure Sub-Sub-TLV is intended for informational use by the control and management planes. It MUST NOT be used at a transit node (as defined in [RFC8754]) for forwarding packets. As an example, this information could be used for:

- o validation of SRv6 SIDs being instantiated in the network and advertised via ISIS. These can be learnt by controllers via BGP-LS and then be monitored for conformance to the SRv6 SID allocation scheme chosen by the operator as described in Section 3.2 of [RFC8986].
- o verification and the automation for securing the SRv6 domain by provisioning filtering rules at SR domain boundaries as described in Section 5 of [RFC8754].

The details of these potential applications are outside the scope of this document.

10. Advertising Endpoint Behaviors

Endpoint behaviors are defined in [RFC8986]. The codepoints for the Endpoint behaviors are defined in the "SRv6 Endpoint Behaviors" registry defined in [RFC8986]. If a behavior is advertised it MUST only be advertised in the TLV[s] marked with "Y" in the table below, and MUST NOT be advertised in the TLV[s] marked with "N" in the table below.

Endpoint Behavior	Endpoint Behavior Codepoint	End SID	End.X SID	Lan End.X SID
End (PSP, USP, USD)	1-4, 28-31	Y	N	N
End.X (PSP, USP, USD)	5-8, 32-35	N	Y	Y
End.DX6	16	N	Y	Y
End.DX4	17	N	Y	Y
End.DT6	18	Y	N	N
End.DT4	19	Y	N	N
End.DT46	20	Y	N	N

11. IANA Considerations

This document requests allocation for the following TLVs, sub-TLVs, and sub-sub-TLVs as well as updating the ISIS TLV registry and defining new registries.

11.1. SRv6 Locator TLV

This document makes the following registrations in the IS-IS TLV Codepoints registry.

Type	Description	IIH	LSP	SNP	Purge
27	SRv6 Locator TLV	n	y	n	n

11.1.1. SRv6 End SID sub-TLV

The SRv6 Locator TLV shares sub-TLV space with TLVs 135, 235, 236 and 237. This document updates the "Sub-TLVs for TLVs 135, 235, 236, and 237 (Extended IP reachability, MT IP. Reach, IPv6 IP. Reach, and MT IPv6 IP. Reach TLVs)" registry defined in [RFC7370]. IANA is requested to update the name of the "Sub-TLVs for TLVs 135, 235, 236, and 237 (Extended IP reachability, MT IP. Reach, IPv6 IP. Reach, and MT IPv6 IP. Reach TLVs)" registry to "Sub-TLVs for TLVs 27, 135, 235, 236, and 237 (SRv6 Locator, Extended IP reachability, MT IP. Reach, IPv6 IP. Reach, and MT IPv6 IP. Reach TLVs)".

IANA is asked to add this document as a reference to (renamed) "Sub-TLVs for TLVs 27, 135, 235, 236, and 237 (SRv6 Locator, Extended IP

reachability, MT IP. Reach, IPv6 IP. Reach, and MT IPv6 IP. Reach TLVs)" registry.

This document makes the following registrations in the (renamed) "Sub-TLVs for TLVs 27, 135, 235, 236, and 237 (SRv6 Locator, Extended IP reachability, MT IP. Reach, IPv6 IP. Reach, and MT IPv6 IP. Reach TLVs)" registry:

Type: 5

Description: SRv6 End SID sub-TLV.

Reference: This document (Section 7.2).

11.1.2. Revised sub-TLV table

The revised table of sub-TLVs for the (renamed) "Sub-TLVs for TLVs 27, 135, 235, 236, and 237 (SRv6 Locator, Extended IP reachability, MT IP. Reach, IPv6 IP. Reach, and MT IPv6 IP. Reach TLVs)" registry is shown below:

Type	27	135	235	236	237
1	y	y	y	y	y
2	y	y	y	y	y
3	n	y	y	y	y
4	y	y	y	y	y
5	y	n	n	n	n
6	n	y	y	y	y
11	y	y	y	y	y
12	y	y	y	y	y
32	n	y	y	y	y

11.2. SRv6 Capabilities sub-TLV

This document makes the following registrations in the "Sub-TLVs for TLV 242 (IS-IS Router CAPABILITY TLV)":

Type: 25

Description: SRv6 Capabilities sub-TLV.

Reference: This document (Section 2).

11.3. Sub-Sub-TLVs of the SRv6 Capability sub-TLV

This document requests a new IANA registry be created under the IS-IS TLV Codepoints Registry to control the assignment of sub-TLV types for the SRv6 Capability sub-TLV specified in this document - Section 2. The suggested name of the new registry is "sub-sub-TLVs of the SRv6 Capability sub-TLV". The registration procedure is "Expert Review" as defined in [RFC8126]. Guidance for the Designated Experts is provided in the [RFC7370]. No sub-sub-TLVs are defined by this document except for the reserved type 0.

Type	Description	Encoding Reference

0	Reserved	
1-255	Unassigned	

11.4. SRv6 End.X SID and SRv6 LAN End.X SID sub-TLVs

This document makes the following registrations in the "Sub-TLVs for TLVs 22, 23, 25, 141, 222, and 223 (Extended IS reachability, IS Neighbor Attribute, L2 Bundle Member Attributes, inter-AS reachability information, MT-ISN, and MT IS Neighbor Attribute TLVs)" registry:

Type: 43

Description: SRv6 End.X SID sub-TLV.

Reference: This document (Section 8.1).

Type: 44

Description: SRv6 LAN End.X SID sub-TLV.

Reference: This document (Section 8.2).

Type 22 23 25 141 222 223

43	y	y	y	y	y	y
44	y	y	y	y	y	y

11.5. MSD Types

This document makes the following registrations in the IGP MSD-Types registry:

Value	Name	Reference
41	SRH Max SL	[This Document]
42	SRH Max End Pop	[This Document]
44	SRH Max H.encaps	[This Document]
45	SRH Max End D	[This Document]

11.6. Sub-Sub-TLVs for SID Sub-TLVs

This document requests a new IANA registry be created under the IS-IS TLV Codepoints Registry to control the assignment of sub-TLV types for the SID Sub-TLVs specified in this document - Section 7.2, Section 8.1, Section 8.2. The suggested name of the new registry is "sub-sub-TLVs for SRv6 End SID and SRv6 End.X SID". The registration procedure is "Expert Review" as defined in [RFC8126]. Guidance for the Designated Experts is provided in [RFC7370]. The following assignments are made by this document:

Type	Description	Encoding Reference
0	Reserved	
1	SRv6 SID Structure Sub-Sub-TLV	[This Document]
2-255	Unassigned	

Type	5	43	44
1	y	y	y

11.7. Prefix Attribute Flags Sub-TLV

This document adds a new bit in the "Bit Values for Prefix Attribute Flags Sub-TLV" registry:

Bit #: 4

Description: Anycast Flag (A-flag)

Reference: This document (Section 6).

11.8. ISIS SRv6 Capabilities sub-TLV Flags Registry

This document requests a new IANA registry be created under the IS-IS TLV Codepoints Registry to control the assignment of bits 0 to 15 in the Flags field of the ISIS SRv6 Capabilities sub-TLV specified in this document (Section 2). The suggested name of the new registry is

"ISIS SRv6 Capabilities sub-TLV Flags". The registration procedure is "Expert Review" as defined in [RFC8126]. Guidance for the Designated Experts is provided in [RFC7370]. The following assignments are made by this document:

Bit #: 1

Description: O-flag

Reference: This document (Section 2).

Bit #: 0, 2-7

Description: Unassigned

11.9. ISIS SRv6 Locator TLV Flags Registry

This document requests a new IANA registry be created under the IS-IS TLV Codepoints Registry to control the assignment of bits 0 to 7 in the Flags field of the ISIS SRv6 Locator TLV specified in this document (Section 7.1). The suggested name of the new registry is "ISIS SRv6 Locator TLV Flags". The registration procedure is "Expert Review" as defined in [RFC8126]. Guidance for the Designated Experts is provided in [RFC7370]. The following assignments are made by this document:

Bit #: 0

Description: D-flag

Reference: This document (Section 7.1).

Bit #: 1-7

Description: Unassigned

11.10. ISIS SRv6 End SID sub-TLV Flags Registry

This document requests a new IANA registry be created under the IS-IS TLV Codepoints Registry to control the assignment of bits 0 to 7 in the Flags field of the ISIS SRv6 End SID sub-TLV specified in this document (Section 7.2). The suggested name of the new registry is "ISIS SRv6 End SID sub-TLV Flags". The registration procedure is "Expert Review" as defined in [RFC8126]. Guidance for the Designated Experts is provided in [RFC7370]. No assignments are made by this document.

Bit #: 0-7

Description: Unassigned

11.11. ISIS SRv6 End.X SID and LAN End.X SID sub-TLVs Flags Registry

This document requests a new IANA registry be created under the IS-IS TLV Codepoints Registry to control the assignment of bits 0 to 7 in the Flags field of the ISIS SRv6 End.X SID and LAN End.X SID sub-TLVs (Section 8.1 and Section 8.2). The suggested name of the new registry is "ISIS SRv6 End.X SID and LAN End.X SID sub-TLVs Flags". The registration procedure is "Expert Review" as defined in [RFC8126]. Guidance for the Designated Experts is provided in [RFC7370]. The following assignments are made by this document:

Bit #: 0

Description: B-flag

Reference: This document (Section 8.1).

Bit #: 1

Description: S-flag

Reference: This document (Section 8.1).

Bit #: 2

Description: P-flag

Reference: This document (Section 8.1).

Bit #: 3-7

Description: Unassigned

12. Security Considerations

Security concerns for IS-IS are addressed in [ISO10589], [RFC5304], and [RFC5310]. While IS-IS is deployed under a single administrative domain, there can be deployments where potential attackers have access to one or more networks in the IS-IS routing domain. In these deployments, the stronger authentication mechanisms defined in the aforementioned documents SHOULD be used.

This document describes the IS-IS extensions required to support Segment Routing over an IPv6 data plane. The security considerations for Segment Routing are discussed in [RFC8402]. [RFC8986] defines the SRv6 Network Programming concept and specifies the main Segment

Routing behaviors to enable the creation of interoperable overlays; the security considerations from that document apply too.

The advertisement for an incorrect MSD value may have negative consequences, see [RFC8491] for additional considerations.

Security concerns associated with the setting of the O-flag are described in [I-D.ietf-6man-spring-srv6-oam].

Security concerns associated with the usage of Flex-Algorithms are described in [I-D.ietf-lsr-flex-algo]).

13. Contributors

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

Stefano Previdi
Huawei Technologies
Email: stefano@previdi.net

Paul Wells
Cisco Systems
Saint Paul,
Minnesota
United States
Email: pauwells@cisco.com

Daniel Voyer
Email: daniel.voyer@bell.ca

Satoru Matsushima
Email: satoru.matsushima@g.softbank.co.jp

Bart Peirens
Email: bart.peirens@proximus.com

Hani Elmalky
Email: hani.elmalky@ericsson.com

Prem Jonnalagadda
Email: prem@barefootnetworks.com

Milad Sharif
Email: msharif@barefootnetworks.com>

Robert Hanzl
Cisco Systems
Millenium Plaza Building, V Celnici 10, Prague 1,
Prague, Czech Republic
Email rhanzl@cisco.com

Ketan Talaulikar
Cisco Systems, Inc.
Email: ketant@cisco.com

14. Acknowledgments

Thanks to Christian Hopps for his review comments and shepherd work.

Thanks to Alvaro Retana and John Scudder for AD review and comments.

15. References

15.1. Normative References

- [I-D.ietf-6man-spring-srv6-oam]
Ali, Z., Filsfils, C., Matsushima, S., Voyer, D., and M. Chen, "Operations, Administration, and Maintenance (OAM) in Segment Routing Networks with IPv6 Data plane (SRv6)", draft-ietf-6man-spring-srv6-oam-11 (work in progress), June 2021.
- [I-D.ietf-lsr-flex-algo]
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-algo-17 (work in progress), July 2021.
- [ISO10589]
International Organization for Standardization, "Intermediate system to Intermediate system intra-domain routing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473)", 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>>.
- [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>>.
- [RFC5308] Hopps, C., "Routing IPv6 with IS-IS", RFC 5308, DOI 10.17487/RFC5308, October 2008, <<https://www.rfc-editor.org/info/rfc5308>>.
- [RFC7370] Ginsberg, L., "Updates to the IS-IS TLV Codepoints Registry", RFC 7370, DOI 10.17487/RFC7370, September 2014, <<https://www.rfc-editor.org/info/rfc7370>>.

- [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>>.
- [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>>.
- [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>>.
- [RFC8491] Tantsura, J., Chunduri, U., Aldrin, S., and L. Ginsberg, "Signaling Maximum SID Depth (MSD) Using IS-IS", RFC 8491, DOI 10.17487/RFC8491, November 2018, <<https://www.rfc-editor.org/info/rfc8491>>.
- [RFC8665] Psenak, P., Ed., Previdi, S., Ed., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", RFC 8665, DOI 10.17487/RFC8665, December 2019, <<https://www.rfc-editor.org/info/rfc8665>>.
- [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>>.
- [RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <<https://www.rfc-editor.org/info/rfc8754>>.

- [RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <<https://www.rfc-editor.org/info/rfc8986>>.

15.2. Informative References

- [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>>.
- [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>>.
- [RFC8355] Filsfils, C., Ed., Previdi, S., Ed., Decraene, B., and R. Shakir, "Resiliency Use Cases in Source Packet Routing in Networking (SPRING) Networks", RFC 8355, DOI 10.17487/RFC8355, March 2018, <<https://www.rfc-editor.org/info/rfc8355>>.

Authors' Addresses

Peter Psenak (editor)
Cisco Systems
Pribinova Street 10
Bratislava 81109
Slovakia

Email: ppsenak@cisco.com

Clarence Filsfils
Cisco Systems
Brussels
Belgium

Email: cfilsfil@cisco.com

Ahmed Bashandy
Individual

Email: abashandy.ietf@gmail.com

Bruno Decraene
Orange
Issy-les-Moulineaux
France

Email: bruno.decraene@orange.com

Zhibo Hu
Huawei Technologies

Email: huzhibo@huawei.com

Internet
Internet-Draft
Intended status: Standards Track
Expires: 13 July 2022

A. Lindem
Cisco Systems
Y. Qu
Futurewei
9 January 2022

OSPF YANG Model Augmentations for Additional Features - Version 1
draft-ietf-lsr-ospf-yang-augmentation-v1-07

Abstract

This document defines YANG data modules augmenting the IETF OSPF YANG model to provide support for Traffic Engineering Extensions to OSPF Version 3 as defined in RF 5329, OSPF Two-Part Metric as defined in RFC 8042, OSPF Graceful Link Shutdown as defined in RFC 8379, OSPF Link-Local Signaling (LLS) Extensions for Local Interface ID Advertisement as defined in RFC 8510, OSPF Application-Specific Link Attributes as defined in RFC 8920, and OSPF Flexible Algorithm.

Status of This Memo

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

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

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

This Internet-Draft will expire on 13 July 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Overview	2
1.1. Requirements Language	3
2. YANG Module for Traffic Engineering Extensions to OSPF Version 3	3
3. YANG Module for OSPF Two-Part Metric	9
4. YANG Module for OSPF Graceful Link Shutdown	13
5. YANG Module for OSPF LLS Extension for Local Interface ID Advertisement	18
6. YANG Module for OSPF Application-Specific Link Attributes	20
7. YANG Module for OSPF Flexible Algorithm	26
8. Security Considerations	46
9. IANA Considerations	47
10. Acknowledgements	48
11. Normative References	48
Authors' Addresses	50

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 YANG data modules augmenting the IETF OSPF YANG model [I-D.ietf-ospf-yang], which itself augments [RFC8349], to provide support for configuration and operational state for the following OSPF features:

RFC5329: Traffic Engineering Extensions to OSPF Version 3 [RFC5329].

RFC8042: OSPF Two-Part Metric [RFC8042].

RFC8379: OSPF Graceful Link Shutdown [RFC8379].

RFC8510: OSPF Link-Local Signaling (LLS) Extensions for Local Interface ID Advertisement [RFC8510].

RFC8920: OSPF Application-Specific Link Attributes [RFC8920].

RFCxxxx: IGP Flexible Algorithm [I-D.ietf-lsr-flex-algo].

The augmentations defined in this document requires support for the OSPF base model[I-D.ietf-ospf-yang] which defines basic OSPF configuration and state. The OSPF YANG model augments the ietf-routing YANG model defined in [RFC8349].

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 [RFC2119].

2. YANG Module for Traffic Engineering Extensions to OSPF Version 3

This document defines a YANG module for Traffic Engineering Extensions to OSPF Version 3 as defined in [RFC5329]. It is an augmentation of the OSPF base model.


```

module: ietf-ospfv3-te
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
  /ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
  /ospf:area-scope-lsa/ospf:version/ospf:ospfv3/ospf:ospfv3
  /ospf:body:
+--ro ospfv3-intra-area-te
+--ro router-address-tlv
  | +--ro router-address?    inet:ipv6-address
+--ro link-tlv
  +--ro link-type                ospf:router-link-type
  +--ro local-if-ipv6-addr
  | +--ro local-if-ipv6-addr*    inet:ipv6-address
+--ro remote-if-ipv6-addr
  | +--ro remote-if-ipv6-addr*  inet:ipv6-address
+--ro te-metric?                uint32
+--ro max-bandwidth?
  | rt-types:bandwidth-ieee-float32
+--ro max-reservable-bandwidth?
  | rt-types:bandwidth-ieee-float32
+--ro unreserved-bandwidths
  | +--ro unreserved-bandwidth*
  |   +--ro priority?          uint8
  |   +--ro unreserved-bandwidth?
  |     rt-types:bandwidth-ieee-float32
+--ro admin-group?             uint32
+--ro neighbor-id
  | +--ro nbr-interface-id     inet:ipv4-address
  | +--ro nbr-router-id       yang:dotted-quad
+--ro unknown-tlvs
  +--ro unknown-tlv*
    +--ro type?               uint16
    +--ro length?             uint16
    +--ro value?              yang:hex-string

<CODE BEGINS> file "ietf-ospfv3-te@2021-07-11.yang"
module ietf-ospfv3-te {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-ospfv3-te";

  prefix ospfv3-te;

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

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

import ietf-ospf {
    prefix "ospf";
}

organization
    "IETF LSR - Link State Routing Working Group";

contact
    "WG Web:    <http://tools.ietf.org/wg/lsr>
    WG List:    <mailto:lsr@ietf.org>

    Author:    Yingzhen Qu
                <mailto:yqu@futurewei.com>
    Author:    Acee Lindem
                <mailto:acee@cisco.com>";

description
    "This YANG module defines the configuration and operational
    state for OSPFv3 extensions to support intra-area Traffic
    Engineering (TE) as defined in RFC 5329.

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

    Copyright (c) 2021 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
```



```
(http://trustee.ietf.org/license-info).

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

reference "RFC XXXX";

revision 2021-07-11 {
  description
    "Initial version";
  reference
    "RFC XXXX: A YANG Data Model for OSPFv3 TE.";
}

identity ospfv3-intra-area-te-lsa {
  base ospf:ospfv3-lsa-type;
  description
    "OSPFv3 intra-area TE LSA.";
}

grouping ospfv3-intra-area-te {
  description "Grouping for OSPFv3 intra-area-te-lsa.";
  container ospfv3-intra-area-te {
    container router-address-tlv {
      description "The router IPv6 address tlv advertises a
        reachable IPv6 address.";
      leaf router-address {
        type inet:ipv6-address;
        description
          "Router IPv6 address.";
      }
    }
  }

  container link-tlv {
    description "Describes a single link, and it is constructed
      of a set of Sub-TLVs.";
    leaf link-type {
      type ospf:router-link-type;
      mandatory true;
      description "Link type.";
    }
  }

  container local-if-ipv6-addr {
    description "All local interface IPv6 addresses.";
    leaf-list local-if-ipv6-addr {
      type inet:ipv6-address;
      description
        "List of local interface IPv6 addresses.";
    }
  }
}
```



```
    }  
  }  
  
  container remote-if-ipv6-addr {  
    description "All remote interface IPv6 addresses.";  
    leaf-list remote-if-ipv6-addr {  
      type inet:ipv6-address;  
      description  
        "List of remote interface IPv6 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.";  
    }  
  }  
  
  leaf admin-group {  
    type uint32;
```



```
        description
          "Administrative group/Resource Class/Color.";
      }

      container neighbor-id {
        description "Neighbor link identification.";
        leaf nbr-interface-id {
          type inet:ipv4-address;
          mandatory true;
          description "The neighbor's interface ID.";
        }
        leaf nbr-router-id {
          type yang:dotted-quad;
          mandatory true;
          description "The neighbor's router ID.";
        }
      }

      uses ospf:unknown-tlvs;
    }

    description "OSPFv3 Intra-Area-TE-LSA.";
    reference "RFC 5329: Traffic Engineering Extensions to OSPF
      :   Version 3.";
  }
}

augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:areas/"
+ "ospf:area/ospf:database/"
+ "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
+ "ospf:area-scope-lsa/ospf:version/ospf:ospfv3/"
+ "ospf:ospfv3/ospf:body" {
  when "../.../.../.../.../.../.../.../"
  + "rt:type = 'ospf:ospfv3'" {
    description
      "This augmentation is only valid for OSPFv3.";
  }
  description
    "OSPFv3 Intra-Area-TE-LSA.";

  uses ospfv3-intra-area-te;
}
}
<CODE ENDS>
```


3. YANG Module for OSPF Two-Part Metric

This document defines a YANG module for OSPF Two-Part Metric feature as defined in [RFC8042]. It is an augmentation of the OSPF base model.

```

module: ietf-ospf-two-part-metric
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
  /ospf:interfaces/ospf:interface:
  +--rw two-part-metric
    +--rw enable?          boolean
    +--rw int-input-cost?  ospf:ospf-link-metric
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
    /ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
    /ospf:area-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
    /ospf:body/ospf:opaque/ospf:extended-link-opaque
    /ospf:extended-link-tlv:
  +--ro network-to-router-metric-sub-tlvs
    +--ro net-to-rtr-sub-tlv*
      +--ro mt-id?          uint8
      +--ro mt-metric?     uint16
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
    /ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
    /ospf:area-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
    /ospf:body/ospf:opaque/ospf:te-opaque/ospf:link-tlv:
  +--ro network-to-router-te-metric?  uint32
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
    /ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
    /ospf:area-scope-lsa/ospf:version/ospf:ospfv3/ospf:ospfv3
    /ospf:body/ospfv3-te:ospfv3-intra-area-te/ospfv3-te:link-tlv:
  +--ro network-to-router-te-metric?  uint32

<CODE BEGINS> file "ietf-ospf-two-part-metric@2021-07-11.yang"
module ietf-ospf-two-part-metric {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-ospf-two-part-metric";

  prefix ospf-two-metric;

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

```



```
import ietf-ospf {
  prefix "ospf";
}

import ietf-ospfv3-te {
  prefix "ospfv3-te";
}

organization
  "IETF LSR - Link State Routing Working Group";

contact
  "WG Web:    <http://tools.ietf.org/wg/lsr>
  WG List:    <mailto:lsr@ietf.org>

  Author:     Yingzhen Qu
               <mailto:yqu@futurewei.com>
  Author:     Acee Lindem
               <mailto:acee@cisco.com>";

description
  "This YANG module defines the configuration and operational
  state for OSPF Two-Part Metric feature as defined in RFC 8042.

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

  Copyright (c) 2021 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
  (http://trustee.ietf.org/license-info).

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

reference "RFC XXXX";

revision 2021-07-11 {
  description
    "Initial version";
  reference
    "RFC XXXX: A YANG Data Model for OSPF.";
}
```



```
identity two-part-metric {
  base ospf:informational-capability;
  description
    "When set, the router is capable of supporting OSPF
    two-part metrics.";
  reference
    "RFC 8042: OSPF Two-Part Metric";
}

/* RFC 8042 */
augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/ospf:ospf/"
  + "ospf:areas/ospf:area/ospf:interfaces/ospf:interface" {
  when ".../.../.../rt:type = 'ospf:ospfv2' or "
  + ".../.../.../rt:type = 'ospf:ospfv3'" {
    description
      "This augments the OSPF interface configuration
      when used.";
  }
  description
    "This augments the OSPF protocol interface
    configuration with two-part metric.";

  container two-part-metric {
    when "enum-value(.../ospf:interface-type) = 2" {
      description
        "Two-part metric when link type is multi-access.";
    }
    leaf enable {
      type boolean;
      default false;
      description
        "Enable two-part metric.";
    }
    leaf int-input-cost {
      type ospf:ospf-link-metric;
      description
        "Link state metric from the two-part-metric network
        to this router.";
    }
    description
      "Interface two part metric configuration.";
  }
}

augment "/rt:routing/"
  + "rt:control-plane-protocols/rt:control-plane-protocol/"
  + "ospf:ospf/ospf:areas/"
```



```

    + "ospf:area/ospf:database/"
    + "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
    + "ospf:area-scope-lsa/ospf:version/ospf:ospfv2/"
    + "ospf:ospfv2/ospf:body/ospf:opaque/"
    + "ospf:extended-link-opaque/ospf:extended-link-tlv" {
when "../.../.../.../.../.../.../.../.../.../.../..."
    + "rt:type = 'ospf:ospfv2'" {
    description
        "This augmentation is only valid for OSPFv2.";
    }
description
    "Network-to-Router metric sub tlv for OSPFv2 extended link TLV
    in type 10 opaque LSA.";

    container network-to-router-metric-sub-tlvs {
        description "Network-to-Router metric sub TLV.";
        list net-to-rtr-sub-tlv {
            leaf mt-id {
                type uint8;
                description "Multi-Topology Identifier (MT-ID).";
            }
            leaf mt-metric {
                type uint16;
                description "Network-to-router metric.";
            }
        }
        description
            "Network-to-Router metric sub-TLV.";
    }
}

augment "/rt:routing/"
    + "rt:control-plane-protocols/rt:control-plane-protocol/"
    + "ospf:ospf/ospf:areas/"
    + "ospf:area/ospf:database/"
    + "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
    + "ospf:area-scope-lsa/ospf:version/ospf:ospfv2/"
    + "ospf:ospfv2/ospf:body/ospf:opaque/ospf:te-opaque/"
    + "ospf:link-tlv" {
when "../.../.../.../.../.../.../.../.../.../.../..."
    + "rt:type = 'ospf:ospfv2'" {
    description
        "This augmentation is only valid for OSPFv2.";
    }
description
    "Traffic Engineering Network-to-Router Sub-TLV.";
    leaf network-to-router-te-metric {
        type uint32;
    }
}

```



```

        description "Network to Router TE metric.";
        reference
            "RFC 8042 - OSPF Two-Part Metric";
    }
}

augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:areas/"
+ "ospf:area/ospf:database/"
+ "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
+ "ospf:area-scope-lsa/ospf:version/ospf:ospfv3/"
+ "ospf:ospfv3/ospf:body/ospfv3-te:ospfv3-intra-area-te/"
+ "ospfv3-te:link-tlv" {
when "../.../.../.../.../.../.../.../.../.../..."
+ "rt:type = 'ospf:ospfv3'" {
    description
        "This augmentation is only valid for OSPFv3.";
}
description
    "Traffic Engineering Network-to-Router Sub-TLV.";
leaf network-to-router-te-metric {
    type uint32;
    description "Network to Router TE metric.";
    reference
        "RFC 8042 - OSPF Two-Part Metric";
}
}
}
<CODE ENDS>

```

4. YANG Module for OSPF Graceful Link Shutdown

This document defines a YANG module for OSPF Graceful Link Shutdown feature as defined in [RFC8379]. It is an augmentation of the OSPF base model.


```

module: ietf-ospf-graceful-link-shutdown
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
  /ospf:interfaces/ospf:interface:
  +--rw graceful-link-shutdown
    +--rw enable?    boolean
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
  /ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
  /ospf:area-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
  /ospf:body/ospf:opaque/ospf:extended-link-opaque
  /ospf:extended-link-tlv:
  +--ro graceful-link-shutdown-sub-tlv!
  +--ro remote-address-sub-tlv
    | +--ro remote-address?    inet:ipv4-address
  +--ro local-remote-int-id-sub-tlv
    +--ro local-int-id?      uint32
    +--ro remote-int-id?    uint32
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
  /ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
  /ospf:area-scope-lsa/ospf:version/ospf:ospfv3/ospf:ospfv3
  /ospf:body/ospfv3-e-lsa:e-router/ospfv3-e-lsa:e-router-tlvs
  /ospfv3-e-lsa:link-tlv:
  +--ro graceful-link-shutdown-sub-tlv!

<CODE BEGINS> file "ietf-ospf-graceful-link-shutdown@2021-07-11.yang"
module ietf-ospf-graceful-link-shutdown {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-ospf-graceful-link-shutdown";

  prefix ospf-grace-linkdown;

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

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

  import ietf-ospf {
    prefix "ospf";
  }

```



```
}

import ietf-ospfv3-extended-lsa {
  prefix "ospfv3-e-lsa";
}

organization
  "IETF LSR - Link State Routing Working Group";

contact
  "WG Web:    <http://tools.ietf.org/wg/lsr>
  WG List:    <mailto:lsr@ietf.org>

  Author:     Yingzhen Qu
               <mailto:yqu@futurewei.com>
  Author:     Acee Lindem
               <mailto:acee@cisco.com>";

description
  "This YANG module defines the configuration and operational
  state for OSPF Graceful Link Shutdown feature as defined
  in RFC 8379.

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

  Copyright (c) 2021 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
  (http://trustee.ietf.org/license-info).

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

reference "RFC XXXX";

revision 2021-07-11 {
  description
    "Initial version";
  reference
    "RFC XXXX: A YANG Data Model for OSPF Graceful Link Shutdown.";
}
```



```
/* RFC 8379 */
augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/ospf:ospf/"
  + "ospf:areas/ospf:area/ospf:interfaces/ospf:interface" {
  when "../.../rt:type = 'ospf:ospfv2' or "
  + ".../rt:type = 'ospf:ospfv3'" {
    description
      "This augments the OSPF interface configuration
      when used.";
  }
  description
    "This augments the OSPF protocol interface
    configuration with segment routing.";

  container graceful-link-shutdown {
    leaf enable {
      type boolean;
      default false;
      description
        "Enable OSPF graceful link shutdown.";
    }
    description
      "OSPF Graceful Link Shutdown.";
  }
}

/* Database */
augment "/rt:routing/"
  + "rt:control-plane-protocols/rt:control-plane-protocol/"
  + "ospf:ospf/ospf:areas/"
  + "ospf:area/ospf:database/"
  + "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
  + "ospf:area-scope-lsa/ospf:version/ospf:ospfv2/"
  + "ospf:ospfv2/ospf:body/ospf:opaque/"
  + "ospf:extended-link-opaque/ospf:extended-link-tlv" {
  when "../.../rt:type = 'ospf:ospfv2'" {
    description
      "This augmentation is only valid for OSPFv2.";
  }
  description
    "OSPF graceful link shutdown for OSPFv2 extended link TLV
    in type 10 opaque LSA.";

  container graceful-link-shutdown-sub-tlv {
    presence "Enable graceful link shutdown";
    description
```



```

        "Graceful-Link-Shutdown sub-TLV identifies the link as being
        gracefully shutdown.";
    }

    container remote-address-sub-tlv {
        leaf remote-address {
            type inet:ipv4-address;
            description
                "Remote IPv4 address used to identify a particular link
                on the remote side.";
        }
        description
            "This sub-TLV specifies the IPv4 address of the remote
            endpoint on the link.";
    }

    container local-remote-int-id-sub-tlv {
        leaf local-int-id {
            type uint32;
            description "Local interface ID.";
        }
        leaf remote-int-id {
            type uint32;
            description "Remote interface ID.";
        }
        description
            "This sub-TLV specifies Local and Remote Interface IDs.";
    }
}

augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:areas/ospf:area/ospf:database/"
+ "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
+ "ospf:area-scope-lsa/ospf:version/ospf:ospfv3/"
+ "ospf:ospfv3/ospf:body/ospfv3-e-lsa:e-router"
+ "/ospfv3-e-lsa:e-router-tlvs/ospfv3-e-lsa:link-tlv" {
    when "'ospf:../../../../../../../../../../../../'"
    + "rt:type' = 'ospf:ospfv3'" {
        description
            "This augmentation is only valid for OSPFv3
            E-Router LSAs";
    }
}

container graceful-link-shutdown-sub-tlv {
    presence "Enable graceful link shutdown";
    description
        "Graceful-Link-Shutdown sub-TLV identifies the link as being
        gracefully shutdown.";
}

```



```
    }
    description
      "Augment OSPFv3 Area scope router-link TLV.";
  }
}
<CODE ENDS>
```

5. YANG Module for OSPF LLS Extension for Local Interface ID Advertisement

This document defines a YANG module for OSPF Link-Local Signaling (LLS) Extensions for Local Interface ID Advertisement feature as defined in [RFC8510]. It is an augmentation of the OSPF base model.

```
module: ietf-ospf-lls-local-id
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/ospf:ospf:
    +--rw lls-int-id
      +--rw enable?   boolean
```

```
<CODE BEGINS> file "ietf-ospf-lls-local-id@2021-07-11.yang"
module ietf-ospf-lls-local-id {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-ospf-lls-local-id";

  prefix ospf-lls-localid;

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

  import ietf-ospf {
    prefix "ospf";
  }

  organization
    "IETF LSR - Link State Routing Working Group";

  contact
    "WG Web:  <http://tools.ietf.org/wg/lsr>
    WG List:  <mailto:lsr@ietf.org>

    Author:   Yingzhen Qu
              <mailto:yqu@futurewei.com>
    Author:   Acee Lindem
```


<mailto:acee@cisco.com>;

description

"This YANG module defines the configuration and operational state for OSPF Link-Local Signaling (LLS) Extensions for Local Interface ID Advertisement feature as defined in RFC 8510.

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

Copyright (c) 2021 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 (<http://trustee.ietf.org/license-info>).

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

reference "RFC XXXX";

revision 2021-07-11 {

description

"Initial version";

reference

"RFC XXXX: A YANG Data Model for OSPF.";

}

augment "/rt:routing/rt:control-plane-protocols"

+ "/rt:control-plane-protocol/ospf:ospf" {

when "../rt:type = 'ospf:ospfv2' or "

+ "../rt:type = 'ospf:ospfv3'" {

description

"This augments the OSPF routing protocol when used.";

}

description

"This augments the OSPF protocol configuration to support LLS extensions for local interface ID advertisement.";

container lls-int-id {

leaf enable {

type boolean;

default false;

description


```

        "Enable LLS to advertise local interface ID.";
    }
    description
        "OSPF LLS Extensions for interface ID.";
    reference "RFC 8510 - OSPF Link-Local Signaling (LLS)
        Extensions for Local Interface ID Advertisement";
    }
}
}
<CODE ENDS>

```

6. YANG Module for OSPF Application-Specific Link Attributes

This document defines a YANG module for OSPF Application-Specific Link Attributes feature as defined in [RFC8920]. It is an augmentation of the OSPF base model.

```

module: ietf-ospf-link-attr
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/ospf:ospf:
    +--rw ospf-link-attr
        +--rw (link-attr-op-mode)
            +--:(legacy)
                | +--rw legacy?          empty
            +--:(transition)
                | +--rw transition?      empty
            +--:(app-specific)
                +--rw app-specific?      empty
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
    /ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
    /ospf:area-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
    /ospf:body/ospf:opaque/ospf:extended-link-opaque
    /ospf:extended-link-tlv:
    +--ro application-specific-link-attributes-sub-tlvs
        +--ro asla-sub-tlvs* []
            +--ro sabm-length?          uint8
            +--ro udabm-length?         uint8
            +--ro sabm
                | +--ro sabm-bits*      identityref
            +--ro udabm
        +--ro link-attributes-sub-sub-tlvs
            +--ro unknown-tlvs
                +--ro unknown-tlv* []
                    +--ro type?          uint16
                    +--ro length?        uint16
                    +--ro value?         yang:hex-string
augment /rt:routing/rt:control-plane-protocols

```



```

    /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
    /ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
    /ospf:area-scope-lsa/ospf:version/ospf:ospfv3/ospf:ospfv3
    /ospf:body/ospfv3-e-lsa:e-router/ospfv3-e-lsa:e-router-tlvs
    /ospfv3-e-lsa:link-tlv:
+--ro application-specific-link-attributes-sub-tlvs
  +--ro asla-sub-tlvs* []
    +--ro sabm-length?          uint8
    +--ro udabm-length?        uint8
    +--ro sabm
    |   +--ro sabm-bits*      identityref
    +--ro udabm
  +--ro link-attributes-sub-sub-tlvs
    +--ro unknown-tlvs
      +--ro unknown-tlv* []
        +--ro type?          uint16
        +--ro length?        uint16
        +--ro value?         yang:hex-string

<CODE BEGINS> file "ietf-ospf-link-attr@2020-10-31.yang"
module ietf-ospf-link-attr {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-ospf-link-attr";

  prefix ospf-link-attr;

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

  import ietf-ospf {
    prefix "ospf";
  }

  import ietf-ospfv3-extended-lsa {
    prefix "ospfv3-e-lsa";
  }

  organization
    "IETF LSR - Link State Routing Working Group";

  contact
    "WG Web:    <http://tools.ietf.org/wg/lsr>
    WG List:    <mailto:lsr@ietf.org>

    Author:     Yingzhen Qu

```



```

    Author: <mailto:yqu@futurewei.com>
           Acee Lindem
           <mailto:acee@cisco.com>
    Author: Stephane Litkowski
           <mailto:slitkows.ietf@gmail.com>;
```

description

"This YANG module defines the configuration and operational state for OSPF application specific link attributes feature as defined in RFC xxxx.

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

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 (<http://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; 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.";

reference "RFC XXXX";

```

revision 2020-10-31 {
  description
    "Initial version";
  reference
    "RFC XXXX: A YANG Data Model for OSPF application specific
    link attributes.";
}
```

```

identity sabm-bit {
  description
    "Base identity for sabm bits.";
}
```



```
identity rsvp-te-bit {
  base sabm-bit;
  description
    "R bit, RSVP-TE.";
}

identity sr-policy-bit {
  base sabm-bit;
  description
    "S bit, Segment Routing Policy.";
}

identity lfa-bit {
  base sabm-bit;
  description
    "F bit, Loop Free Alternate (LFA). Includes all LFA types.";
}

grouping application-specific-link-attributes-sub-tlvs {
  description
    "OSPF Application-Specific Link Attributes (ASLA) sub-TLV.
    The ASLA sub-TLV is a sub-TLV of the OSPFv2 Extended Link
    TLV [RFC7684] and OSPFv3 Router-Link TLV [RFC8362].";

  container application-specific-link-attributes-sub-tlvs {
    description "Application-Specific Link Attributes sub-TLV.";
    list asla-sub-tlvs {
      leaf sabm-length {
        type uint8;
        description
          "Standard Application Identifier Bit Mask Length in
          octets.";
      }
      leaf udabm-length {
        type uint8;
        description
          "User Defined Application Identifier Bit Mask Length
          in octets.";
      }
    }
    container sabm {
      leaf-list sabm-bits {
        type identityref {
          base sabm-bit;
        }
      }
      description
        "SABM bits list. This list will contain
        identities for the bits which are set in the
        SABA bits.";
    }
  }
}
```



```
    }
    description
      "Standard Application Identifier Bit Mask.";
  }
  container udabm {
    description
      "User Defined Application Identifier Bit Mask.
      This container is to be augmented by user defined
      applications.";
  }
  container link-attributes-sub-sub-tlvs {
    uses ospf:unknown-tlvs;
    description
      "Link Attributes sub-sub-TLVs.";
  }
  description
    "List of application-Specific Link Attributes sub-TLVs.";
}
}

augment "/rt:routing/rt:control-plane-protocols"
+ "/rt:control-plane-protocol/ospf:ospf" {
  when "../rt:type = 'ospf:ospfv2' or "
  + "../rt:type = 'ospf:ospfv3'" {
    description
      "This augments the OSPF routing protocol when used.";
  }
  description
    "This augments OSPF protocol configuration
    with application-specific link attributes.";

  container ospf-link-attr {
    choice link-attr-op-mode {
      mandatory "true";
      leaf legacy {
        type empty;
        description
          "Only send legacy advertisements.";
      }
      leaf transition {
        type empty;
        description
          "Send both application-specific and legacy
          advertisements.";
      }
    }
    leaf app-specific {
      type empty;
    }
  }
}
```



```

        description
            "Only send application-specific advertisements.";
    }
    description
        "Link attributes mode";
    }
    description
        "Link attributes operation mode.";
    }
}

/* Database */
augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:areas/"
+ "ospf:area/ospf:database/"
+ "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
+ "ospf:area-scope-lsa/ospf:version/ospf:ospfv2/"
+ "ospf:ospfv2/ospf:body/ospf:opaque/"
+ "ospf:extended-link-opaque/ospf:extended-link-tlv" {
when "../.../.../.../.../.../.../.../.../.../..."
+ "rt:type = 'ospf:ospfv2'" {
    description
        "This augmentation is only valid for OSPFv2.";
}
description
    "OSPF Application-Specific Link Attributes (ASLA) sub-TLV is
    a sub-TLV of OSPFv2 Extended Link TLV (RFC7684).";

    uses application-specific-link-attributes-sub-tlvs;
}

augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:areas/ospf:area/ospf:database/"
+ "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
+ "ospf:area-scope-lsa/ospf:version/ospf:ospfv3/"
+ "ospf:ospfv3/ospf:body/ospfv3-e-lsa:e-router"
+ "/ospfv3-e-lsa:e-router-tlvs/ospfv3-e-lsa:link-tlv" {
when "'ospf:../.../.../.../.../.../.../.../.../.../..."
+ "rt:type' = 'ospf:ospfv3'" {
    description
        "This augmentation is only valid for OSPFv3
        E-Router LSAs";
}
description
    "Augment OSPFv3 Area scope router-link TLV.";

```



```

    uses application-specific-link-attributes-sub-tlvs;
  }
}
<CODE ENDS>

```

7. YANG Module for OSPF Flexible Algorithm

This document defines a YANG module for OSPF Flexible Algorithm as defined in [I-D.ietf-lsr-flex-algo]. It is an augmentation of the OSPF base model.

```

module: ietf-ospf-flex-algo
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/ospf:ospf:
    +--rw ospf-flex-algo
      +--rw flex-algo* [algo-number]
        +--rw algo-number          uint8
        +--rw advertise-definition? boolean
        +--rw admin-groups {te-types:extended-admin-groups,
                           te-types:named-extended-admin-groups}?
          | +--rw exclude-admin-groups* -> /te:te/globals
          | | /named-admin-groups
          | | /named-admin-group/name
          | +--rw include-any-admin-groups* -> /te:te/globals
          | | /named-admin-groups
          | | /named-admin-group/name
          | +--rw include-all-admin-groups* -> /te:te/globals
          | | /named-admin-groups
          | | /named-admin-group/name
          +--rw exclude-srlgs*      -> /te:te/globals/named-srlgs
          | /named-srlg/name
          | {te-types:named-srlg-groups}?
          +--rw fast-reroute?        boolean
          +--rw metric-type?         identityref
          +--rw microloop-avoidance? boolean
          +--rw prefix-metric!
          +--rw priority?            uint8
      augment /rt:routing/rt:control-plane-protocols
        /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
        /ospf:interfaces/ospf:interface/ospf:database
        /ospf:link-scope-lsa-type/ospf:link-scope-lsas
        /ospf:link-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
        /ospf:body/ospf:opaque/ospf:ri-opaque:
          +--ro fad-tlvs
            +--ro fad-tlv* []
              +--ro flex-algo?      uint8
              +--ro metric-type?    identityref
              +--ro calc-type?      uint8

```



```

    +--ro priority?                uint8
    +--ro fa-ex-ag-sub-tlv
    |   +--ro extended-admin-groups*  uint64
    +--ro fa-in-any-ag-sub-tlv
    |   +--ro extended-admin-groups*  uint64
    +--ro fa-in-all-ag-sub-tlv
    |   +--ro extended-admin-groups*  uint64
    +--ro fad-flags-sub-tlv
    |   +--ro fad-flags*  identityref
    +--ro fa-ex-srlg-sub-tlv
    |   +--ro srlgs*  uint32
    +--ro unknown-tlvs
    |   +--ro unknown-tlv*  []
    |   |   +--ro type?  uint16
    |   |   +--ro length?  uint16
    |   |   +--ro value?  yang:hex-string
augment /rt:routing/rt:control-plane-protocols
/rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
/ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
/ospf:area-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
/ospf:body/ospf:opaque/ospf:ri-opaque:
+--ro fad-tlvs
|   +--ro fad-tlv*  []
|   |   +--ro flex-algo?  uint8
|   |   +--ro metric-type?  identityref
|   |   +--ro calc-type?  uint8
|   |   +--ro priority?  uint8
|   |   +--ro fa-ex-ag-sub-tlv
|   |   |   +--ro extended-admin-groups*  uint64
|   |   +--ro fa-in-any-ag-sub-tlv
|   |   |   +--ro extended-admin-groups*  uint64
|   |   +--ro fa-in-all-ag-sub-tlv
|   |   |   +--ro extended-admin-groups*  uint64
|   |   +--ro fad-flags-sub-tlv
|   |   |   +--ro fad-flags*  identityref
|   |   +--ro fa-ex-srlg-sub-tlv
|   |   |   +--ro srlgs*  uint32
|   |   +--ro unknown-tlvs
|   |   |   +--ro unknown-tlv*  []
|   |   |   |   +--ro type?  uint16
|   |   |   |   +--ro length?  uint16
|   |   |   |   +--ro value?  yang:hex-string
augment /rt:routing/rt:control-plane-protocols
/rt:control-plane-protocol/ospf:ospf/ospf:database
/ospf:as-scope-lsa-type/ospf:as-scope-lsas/ospf:as-scope-lsa
/ospf:version/ospf:ospfv2/ospf:ospfv2/ospf:body/ospf:opaque
/ospf:ri-opaque:
+--ro fad-tlvs

```



```

    +---ro fad-tlv* []
      +---ro flex-algo?          uint8
      +---ro metric-type?       identityref
      +---ro calc-type?         uint8
      +---ro priority?          uint8
      +---ro fa-ex-ag-sub-tlv
      | +---ro extended-admin-groups*  uint64
      +---ro fa-in-any-ag-sub-tlv
      | +---ro extended-admin-groups*  uint64
      +---ro fa-in-all-ag-sub-tlv
      | +---ro extended-admin-groups*  uint64
      +---ro fad-flags-sub-tlv
      | +---ro fad-flags*    identityref
      +---ro fa-ex-srlg-sub-tlv
      | +---ro srlgs*    uint32
      +---ro unknown-tlvs
      | +---ro unknown-tlv* []
      |   +---ro type?    uint16
      |   +---ro length?  uint16
      |   +---ro value?   yang:hex-string
augment /rt:routing/rt:control-plane-protocols
/rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
/ospf:interfaces/ospf:interface/ospf:database
/ospf:link-scope-lsa-type/ospf:link-scope-lsas
/ospf:link-scope-lsa/ospf:version/ospf:ospfv3/ospf:ospfv3
/ospf:body/ospf:router-information:
+---ro fad-tlvs
  +---ro fad-tlv* []
    +---ro flex-algo?          uint8
    +---ro metric-type?       identityref
    +---ro calc-type?         uint8
    +---ro priority?          uint8
    +---ro fa-ex-ag-sub-tlv
    | +---ro extended-admin-groups*  uint64
    +---ro fa-in-any-ag-sub-tlv
    | +---ro extended-admin-groups*  uint64
    +---ro fa-in-all-ag-sub-tlv
    | +---ro extended-admin-groups*  uint64
    +---ro fad-flags-sub-tlv
    | +---ro fad-flags*    identityref
    +---ro fa-ex-srlg-sub-tlv
    | +---ro srlgs*    uint32
    +---ro unknown-tlvs
    | +---ro unknown-tlv* []
    |   +---ro type?    uint16
    |   +---ro length?  uint16
    |   +---ro value?   yang:hex-string
augment /rt:routing/rt:control-plane-protocols

```



```

    /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
    /ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
    /ospf:area-scope-lsa/ospf:version/ospf:ospfv3/ospf:ospfv3
    /ospf:body/ospf:router-information:
+--ro fad-tlvs
  +--ro fad-tlv* []
    +--ro flex-algo?          uint8
    +--ro metric-type?        identityref
    +--ro calc-type?          uint8
    +--ro priority?           uint8
    +--ro fa-ex-ag-sub-tlv
      | +--ro extended-admin-groups*  uint64
    +--ro fa-in-any-ag-sub-tlv
      | +--ro extended-admin-groups*  uint64
    +--ro fa-in-all-ag-sub-tlv
      | +--ro extended-admin-groups*  uint64
    +--ro fad-flags-sub-tlv
      | +--ro fad-flags*  identityref
    +--ro fa-ex-srlg-sub-tlv
      | +--ro srlgs*  uint32
    +--ro unknown-tlvs
      +--ro unknown-tlv* []
        +--ro type?  uint16
        +--ro length?  uint16
        +--ro value?  yang:hex-string
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/ospf:ospf/ospf:database
    /ospf:as-scope-lsa-type/ospf:as-scope-lsas/ospf:as-scope-lsa
    /ospf:version/ospf:ospfv3/ospf:ospfv3/ospf:body
    /ospf:router-information:
+--ro fad-tlvs
  +--ro fad-tlv* []
    +--ro flex-algo?          uint8
    +--ro metric-type?        identityref
    +--ro calc-type?          uint8
    +--ro priority?           uint8
    +--ro fa-ex-ag-sub-tlv
      | +--ro extended-admin-groups*  uint64
    +--ro fa-in-any-ag-sub-tlv
      | +--ro extended-admin-groups*  uint64
    +--ro fa-in-all-ag-sub-tlv
      | +--ro extended-admin-groups*  uint64
    +--ro fad-flags-sub-tlv
      | +--ro fad-flags*  identityref
    +--ro fa-ex-srlg-sub-tlv
      | +--ro srlgs*  uint32
    +--ro unknown-tlvs
      +--ro unknown-tlv* []

```



```

        +--ro type?          uint16
        +--ro length?        uint16
        +--ro value?         yang:hex-string
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
  /ospf:interfaces/ospf:interface/ospf:database
  /ospf:link-scope-lsa-type/ospf:link-scope-lsas
  /ospf:link-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
  /ospf:body/ospf:opaque/ospf:extended-prefix-opaque
  /ospf:extended-prefix-tlv:
+--ro fapm-sub-tlvs
  +--ro fapm-sub-tlv* []
    +--ro flex-algo?      uint8
    +--ro fapm-flags*     identityref
    +--ro metric?         uint32
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
  /ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
  /ospf:area-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
  /ospf:body/ospf:opaque/ospf:extended-prefix-opaque
  /ospf:extended-prefix-tlv:
+--ro fapm-sub-tlvs
  +--ro fapm-sub-tlv* []
    +--ro flex-algo?      uint8
    +--ro fapm-flags*     identityref
    +--ro metric?         uint32
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/ospf:ospf/ospf:database
  /ospf:as-scope-lsa-type/ospf:as-scope-lsas/ospf:as-scope-lsa
  /ospf:version/ospf:ospfv2/ospf:ospfv2/ospf:body/ospf:opaque
  /ospf:extended-prefix-opaque/ospf:extended-prefix-tlv:
+--ro fapm-sub-tlvs
  +--ro fapm-sub-tlv* []
    +--ro flex-algo?      uint8
    +--ro fapm-flags*     identityref
    +--ro metric?         uint32
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
  /ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
  /ospf:area-scope-lsa/ospf:version/ospf:ospfv3/ospf:ospfv3
  /ospf:body/ospfv3-e-lsa:e-inter-area-prefix
  /ospfv3-e-lsa:e-inter-prefix-tlvs
  /ospfv3-e-lsa:inter-prefix-tlv:
+--ro fapm-sub-tlvs
  +--ro fapm-sub-tlv* []
    +--ro flex-algo?      uint8
    +--ro fapm-flags*     identityref
    +--ro metric?         uint32

```



```

augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/ospf:ospf/ospf:database
  /ospf:as-scope-lsa-type/ospf:as-scope-lsas/ospf:as-scope-lsa
  /ospf:version/ospf:ospfv3/ospf:ospfv3/ospf:body
  /ospfv3-e-lsa:e-as-external/ospfv3-e-lsa:e-external-tlvs
  /ospfv3-e-lsa:external-prefix-tlv:
  +--ro fapm-sub-tlvs
    +--ro fapm-sub-tlv* []
      +--ro flex-algo?      uint8
      +--ro fapm-flags*    identityref
      +--ro metric?        uint32
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
  /ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
  /ospf:area-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
  /ospf:body/ospf:opaque:
  +--ro eia-asbr-opaque
    +--ro eia-asbr-tlv
      +--ro asbr-rtr-id?    rt-types:router-id
      +--ro faam-sub-tlvs
        +--ro faam-sub-tlv* []
          +--ro flex-algo?    uint8
          +--ro metric?      uint32
      +--ro unknown-tlvs
        +--ro unknown-tlv* []
          +--ro type?        uint16
          +--ro length?      uint16
          +--ro value?       yang:hex-string
    +--ro unknown-tlvs
      +--ro unknown-tlv* []
        +--ro type?        uint16
        +--ro length?      uint16
        +--ro value?       yang:hex-string
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
  /ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
  /ospf:area-scope-lsa/ospf:version/ospf:ospfv3/ospf:ospfv3
  /ospf:body/ospfv3-e-lsa:e-inter-area-router
  /ospfv3-e-lsa:e-inter-router-tlvs
  /ospfv3-e-lsa:inter-router-tlv:
  +--ro faam-sub-tlvs
    +--ro faam-sub-tlv* []
      +--ro flex-algo?    uint8
      +--ro metric?      uint32

```



```
<CODE BEGINS> file "ietf-ospf-flex-algo@2021-06-18.yang"
module ietf-ospf-flex-algo {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-ospf-flex-algo";
  prefix ospf-flex-algo;

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

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

  import ietf-te-types {
    prefix te-types;
    reference
      "RFC8776: Common YANG Data Types for Traffic Engineering.";
  }

  import ietf-ospf {
    prefix "ospf";
  }

  import ietf-ospfv3-extended-lsa {
    prefix "ospfv3-e-lsa";
  }

  import ietf-te {
    prefix "te";
  }

  organization
    "IETF LSR - Link State Routing Working Group";
  contact
    "WG Web:  <https://tools.ietf.org/wg/spring/>
    WG List:  <mailto:spring@ietf.org>

    Author:   Yingzhen Qu
              <mailto:yingzhen.qu@futurewei.com>
    Author:   Acee Lindem
              <mailto:acee@cisco.com>
    Author:   Stephane Litkowski
```



```
        <mailto:slitkows.ietf@gmail.com>
    ";

description
    "The YANG module defines the configuration and operational
    state for OSPF Flexible Algorithm as defined in RFC xxxx.

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

    Copyright (c) 2021 IETF Trust and the persons identified as
    authors of the code. All rights reserved.

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

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

    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."

reference "RFC XXXX: YANG Data Model for OSPF Flexible Algorithm.";

revision 2021-06-18 {
    description
        "Initial Version";
    reference "RFC XXXX: YANG Data Model for OSPF Flexible Algorithm.";
}

/* Identities */

identity metric-type {
    description
        "Base identity for route metric types.";
}

identity igp-metric {
    base metric-type;
    description
```



```
    "Identity for the IGP metric type.";
}

identity min-uni-delay {
    base metric-type;
    description
        "Min unidirectional link delay metric type.";
    reference
        "RFC 7471 - OSPF Traffic Engineering (TE) Metric Extensions";
}

identity te-metric {
    base metric-type;
    description
        "Traffic engineering metric type.";
    reference
        "RFC 3630 - Traffic Engineering (TE) Extensions to OSPF
        Version 2";
}

identity fad-flags {
    description
        "Base identity for OSPF FAD flags.";
}

identity m-bit {
    base fad-flags;
    description
        "M bit, when set, the fex-algo specific prefix and ASBR
        metric MUST be used for inter-area and external prefix
        calculation.";
}

identity fapm-flags {
    description
        "Base identity for OSPF flex-algo prefix metric flags.";
}

identity e-bit {
    base fapm-flags;
    description
        "External metric, if set, the metric specified is a
        type 2 external metric.";
}

/* Groupings */
grouping fa-ex-ag-sub-tlv {
    container fa-ex-ag-sub-tlv {
```



```
    leaf-list extended-admin-groups {
      type uint64;
      description
        "Extended administrative group as defined in RFC 7308.";
    }
    description
      "The flex-algo exclude admin group sub-tlv.";
  }
  description
    "The flex-algo exclude admin group sub-tlv.";
}

grouping fa-in-any-ag-sub-tlv {
  container fa-in-any-ag-sub-tlv {
    leaf-list extended-admin-groups {
      type uint64;
      description
        "Extended administrative group as defined in RFC 7308.";
    }
    description
      "The flex-algo include-any admin group sub-tlv.";
  }
  description
    "The flex-algo include-any admin group sub-tlv.";
}

grouping fa-in-all-ag-sub-tlv {
  container fa-in-all-ag-sub-tlv {
    leaf-list extended-admin-groups {
      type uint64;
      description
        "Extended administrative group as defined in RFC 7308.";
    }
    description
      "The flex-algo include-all admin group sub-tlv.";
  }
  description
    "The flex-algo include-all admin group sub-tlv.";
}

grouping fad-flags-sub-tlv {
  container fad-flags-sub-tlv {
    leaf-list fad-flags {
      type identityref {
        base fad-flags;
      }
    }
    description
      "Flex-algo definition flags list.";
  }
}
```



```
    }
    description
      "OSPF flex-algo definition flags.";
  }
  description
    "The flex-algo definition flags sub-tlv.";
}

grouping fa-ex-srlg-sub-tlv {
  container fa-ex-srlg-sub-tlv {
    leaf-list srlgs {
      type uint32;
      description
        "SRLG value as defined in RFC 4203.";
    }
    description
      "The flex-algo exclude SRLG sub-tlv.";
  }
  description
    "The flex-algo exclude SRLG sub-tlv.";
}

grouping fad-tlvs {
  container fad-tlvs {
    list fad-tlv {
      leaf flex-algo {
        type uint8;
        description
          "Flex-algo number, value between 128 and 255 inclusive.";
      }
      leaf metric-type {
        type identityref {
          base metric-type;
        }
        description
          "Type of metric to be used during the calculation.";
      }
      leaf calc-type {
        type uint8 {
          range "0..127";
        }
        description
          "IGP algorithm types, value from 0 to 127 as
          defined under 'Interior Gateway Protocol (IGP)
          Parameter' by IANA.";
      }
      leaf priority {
        type uint8;
      }
    }
  }
}
```



```
        description
            "Priority of the advertisement.";
    }

    uses fa-ex-ag-sub-tlv;
    uses fa-in-any-ag-sub-tlv;
    uses fa-in-all-ag-sub-tlv;
    uses fad-flags-sub-tlv;
    uses fa-ex-srlg-sub-tlv;
    uses ospf:unknown-tlvs;

    description
        "List of flex-algo definition TLVs.";
    }
    description
        "OSPF Flexible Algorithm Definition TLV.";
    }
    description
        "OSPF Flexible Algorithm Definition (FAD) TLV.";
    }

    grouping fapm-sub-tlvs {
        container fapm-sub-tlvs {
            list fapm-sub-tlv {
                leaf flex-algo {
                    type uint8;
                    description
                        "Flex-algo number, value between 128 and 255
                        inclusive.";
                }
                leaf-list fapm-flags {
                    type identityref {
                        base fapm-flags;
                    }
                    description
                        "Flex-algo prefix metric flags list.";
                }
                leaf metric {
                    type uint32;
                    description
                        "Prefix metric.";
                }
            }
            description
                "List of flex-algo prefix sub-tlvs.";
        }
        description
            "Flex-algo prefix metric sub-tlvs.";
    }
}
```



```
    description
      "Flexible Algorithm Prefix Metric (FAPM) sub TLVs.";
  }

  grouping faam-sub-tlvs {
    container faam-sub-tlvs {
      list faam-sub-tlv {
        leaf flex-algo {
          type uint8;
          description
            "Flex-algo number, value between 128 and 255
            inclusive.";
        }
        leaf metric {
          type uint32;
          description
            "Prefix metric.";
        }
        description
          "List of faam sub-tlvs.";
      }
      description
        "Flexible Algorithm ASBR Metric (FAAM) Sub-TLVs.";
    }
    description
      "Flexible Algorithm ASBR Metric (FAAM) Sub-TLVs.";
  }

  /* Configurations */

  augment "/rt:routing/rt:control-plane-protocols"
    + "/rt:control-plane-protocol/ospf:ospf" {
    when "../rt:type = 'ospf:ospfv2' or "
      + "../rt:type = 'ospf:ospfv3'" {
      description
        "This augments the OSPF routing protocol when used.";
    }
    description
      "This augments OSPF protocol with Flexible
      Algorithm.";

    container ospf-flex-algo {
      list flex-algo {
        key "algo-number";

        leaf algo-number {
          type uint8 {
```



```
        range "128..255";
    }
    description
        "An identifier in the range 128-255 that's associated
        with the Flexible Algorithm Definition.";
}

leaf advertise-definition {
    type boolean;
    default true;
    description
        "Enable to advertise the flex-algo definition.";
}

container admin-groups {
    if-feature "te-types:extended-admin-groups";
    if-feature "te-types:named-extended-admin-groups";
    leaf-list exclude-admin-groups {
        type leafref {
            path "/te:te/te:globals/te:named-admin-groups/"
                + "te:named-admin-group/te:name";
        }
        description
            "Exclude rule used during the flex-algo
            path computation.";
    }
    leaf-list include-any-admin-groups {
        type leafref {
            path "/te:te/te:globals/te:named-admin-groups/"
                + "te:named-admin-group/te:name";
        }
        description
            "Include-any rule used during the flex-algo
            path computation.";
    }
    leaf-list include-all-admin-groups {
        type leafref {
            path "/te:te/te:globals/te:named-admin-groups/"
                + "te:named-admin-group/te:name";
        }
        description
            "Include-all rule used during the flex-algo
            path computation.";
    }
    description
        "Specify links for the flex-algo path computation.";
}
```



```
leaf-list exclude-srlgs {
  if-feature "te-types:named-srlg-groups";
  type leafref {
    path "/te:te/te:globals/te:named-srlgs/te:named-srlg/"
      + "te:name";
  }
  description
    "Shared Risk Link Groups (SRLGs) to be excluded during
    the flex-algo path computation.";
}

leaf fast-reroute {
  type boolean;
  default true;
  description
    "Enable fast reroute.";
}

leaf metric-type {
  type identityref {
    base metric-type;
  }
  description
    "Type of metric to be used during the calculation.";
}

leaf microloop-avoidance {
  type boolean;
  default true;
  description
    "Enable microloop avoidance.";
}

container prefix-metric {
  presence
    "Use flex-algo specific prefix metric.";
  description
    "Use flex-algo prefix metric.";
}

leaf priority {
  type uint8;
  description
    "Priority of the advertisement.";
}

description
  "List of flex-algo configurations.";
```



```

    }
    description
      "Flexible Algorithm configuration.";
  }
}

/* Database */

augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:areas/ospf:area/"
+ "ospf:interfaces/ospf:interface/ospf:database/"
+ "ospf:link-scope-lsa-type/ospf:link-scope-lsas/"
+ "ospf:link-scope-lsa/ospf:version/ospf:ospfv2/"
+ "ospf:ospfv2/ospf:body/ospf:opaque/ospf:ri-opaque" {
  when "../.../.../.../.../.../.../.../.../.../..."
  + "rt:type = 'ospf:ospfv2'" {
    description
      "This augmentation is only valid for OSPFv2.";
  }

  description
    "Flex-algo definition TLVs for OSPFv2 type 9 opaque RI LSA.";

  uses fad-tlvs;
}

augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:areas/"
+ "ospf:area/ospf:database/"
+ "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
+ "ospf:area-scope-lsa/ospf:version/ospf:ospfv2/"
+ "ospf:ospfv2/ospf:body/ospf:opaque/ospf:ri-opaque" {
  when "../.../.../.../.../.../.../.../.../.../..."
  + "rt:type = 'ospf:ospfv2'" {
    description
      "This augmentation is only valid for OSPFv2.";
  }

  description
    "Flex-algo definition TLVs for OSPFv2 type 10 opaque RI LSA.";

  uses fad-tlvs;
}

augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"

```



```

    + "ospf:ospf/ospf:database/"
    + "ospf:as-scope-lsa-type/ospf:as-scope-lsas/"
    + "ospf:as-scope-lsa/ospf:version/ospf:ospfv2/"
    + "ospf:ospfv2/ospf:body/ospf:opaque/ospf:ri-opaque" {
when "../.../.../.../.../.../.../.../..."
    + "rt:type = 'ospf:ospfv2'" {
    description
        "This augmentation is only valid for OSPFv2.";
    }
    description
        "Flex-algo definition TLVs for OSPFv2 type 11 opaque RI LSA.";

    uses fad-tlvs;
}

augment "/rt:routing/"
    + "rt:control-plane-protocols/rt:control-plane-protocol/"
    + "ospf:ospf/ospf:areas/ospf:area/"
    + "ospf:interfaces/ospf:interface/ospf:database/"
    + "ospf:link-scope-lsa-type/ospf:link-scope-lsas/"
    + "ospf:link-scope-lsa/ospf:version/ospf:ospfv3/"
    + "ospf:ospfv3/ospf:body/ospf:router-information" {
when "../.../.../.../.../.../.../.../..."
    + "rt:type = 'ospf:ospfv3'" {
    description
        "This augmentation is only valid for OSPFv3.";
    }

    description
        "Flex-algo definition TLVs for OSPFv3 Router Information LSA.";

    uses fad-tlvs;
}

augment "/rt:routing/"
    + "rt:control-plane-protocols/rt:control-plane-protocol/"
    + "ospf:ospf/ospf:areas/"
    + "ospf:area/ospf:database/"
    + "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
    + "ospf:area-scope-lsa/ospf:version/ospf:ospfv3/"
    + "ospf:ospfv3/ospf:body/ospf:router-information" {
when "../.../.../.../.../.../.../.../..."
    + "rt:type = 'ospf:ospfv3'" {
    description
        "This augmentation is only valid for OSPFv3.";
    }

    description

```



```

    "Flex-algo definition TLVs for OSPFv3 Router Information LSA.";

    uses fad-tlvs;
}

augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:database/"
+ "ospf:as-scope-lsa-type/ospf:as-scope-lsas/"
+ "ospf:as-scope-lsa/ospf:version/ospf:ospfv3/"
+ "ospf:ospfv3/ospf:body/ospf:router-information" {
when "../.../.../.../.../.../.../..."
+ "rt:type = 'ospf:ospfv3'" {
    description
        "This augmentation is only valid for OSPFv3.";
}
description
    "Flex-algo definition TLVs for OSPFv3 Router Information LSA.";

    uses fad-tlvs;
}

augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:areas/ospf:area/"
+ "ospf:interfaces/ospf:interface/ospf:database/"
+ "ospf:link-scope-lsa-type/ospf:link-scope-lsas/"
+ "ospf:link-scope-lsa/ospf:version/ospf:ospfv2/"
+ "ospf:ospfv2/ospf:body/ospf:opaque/"
+ "ospf:extended-prefix-opaque/ospf:extended-prefix-tlv" {
when "../.../.../.../.../.../.../.../.../.../.../..."
+ "rt:type = 'ospf:ospfv2'" {
    description
        "This augmentation is only valid for OSPFv2.";
}
description
    "Flex-algo prefix metric sub TLVs for OSPFv2 extended prefix TLV
    in type 9 opaque LSA.";
    uses fapm-sub-tlvs;
}

augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:areas/"
+ "ospf:area/ospf:database/"
+ "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
+ "ospf:area-scope-lsa/ospf:version/ospf:ospfv2/"
+ "ospf:ospfv2/ospf:body/ospf:opaque/"

```



```

    + "ospf:extended-prefix-opaque/ospf:extended-prefix-tlv" {
when "../../../../../../../../../../../../../../../"
    + "rt:type = 'ospf:ospfv2'" {
    description
        "This augmentation is only valid for OSPFv2.";
    }
    description
        "Flex-algo prefix metric sub TLVs for OSPFv2 extended prefix TLV
        in type 10 opaque LSA.";
    uses fapm-sub-tlvs;
}

augment "/rt:routing/"
    + "rt:control-plane-protocols/rt:control-plane-protocol/"
    + "ospf:ospf/ospf:database/"
    + "ospf:as-scope-lsa-type/ospf:as-scope-lsas/"
    + "ospf:as-scope-lsa/ospf:version/ospf:ospfv2/"
    + "ospf:ospfv2/ospf:body/ospf:opaque/"
    + "ospf:extended-prefix-opaque/ospf:extended-prefix-tlv" {
when "../../../../../../../../../../../"
    + "rt:type = 'ospf:ospfv2'" {
    description
        "This augmentation is only valid for OSPFv2.";
    }
    description
        "Flex-algo prefix metric sub TLVs for OSPFv2 extended prefix TLV
        in type 11 opaque LSA.";
    uses fapm-sub-tlvs;
}

/* Flex-algo prefix metric Sub-TLV in OSPFv3 Inter-Area Prefix TLV */
augment "/rt:routing/"
    + "rt:control-plane-protocols/rt:control-plane-protocol/"
    + "ospf:ospf/ospf:areas/ospf:area/ospf:database/"
    + "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
    + "ospf:area-scope-lsa/ospf:version/ospf:ospfv3/"
    + "ospf:ospfv3/ospf:body/ospfv3-e-lsa:e-inter-area-prefix/"
    + "ospfv3-e-lsa:e-inter-prefix-tlvs/ospfv3-e-lsa:inter-prefix-tlv" {
when "../../../../../../../../../../../"
    + "rt:type = 'ospf:ospfv3'" {
    description
        "This augmentation is only valid for OSPFv3
        E-Router LSAs";
    }
    uses fapm-sub-tlvs;
    description
        "OSPFv3 Area-Scoped Inter-Area Prefix TLV.";
}

```



```

/* Flex-algo prefix metric Sub-TLV in OSPFv3 External Prefix TLV */
augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:database/"
+ "ospf:as-scope-lsa-type/ospf:as-scope-lsas/"
+ "ospf:as-scope-lsa/ospf:version/ospf:ospfv3/"
+ "ospf:ospfv3/ospf:body/ospfv3-e-lsa:e-as-external/"
+ "ospfv3-e-lsa:e-external-tlvs/ospfv3-e-lsa:external-prefix-tlv" {
when "'ospf:../../../../../../../../../../../../'"
+ "rt:type" = 'ospf:ospfv3' {
description
    "This augmentation is only valid for OSPFv3.";
}
uses fapm-sub-tlvs;
description
    "OSPFv3 AS-Scoped External Prefix TLV.";
}

/* OSPFv2 Extended Inter-Area ASBR LSA */
augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:areas/"
+ "ospf:area/ospf:database/"
+ "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
+ "ospf:area-scope-lsa/ospf:version/ospf:ospfv2/"
+ "ospf:ospfv2/ospf:body/ospf:opaque" {
when " '../../../../../../../../../../../../'"
+ "rt:type" = 'ospf:ospfv2' {
description
    "This augmentation is only valid for OSPFv2.";
}
description
    "OSPFv2 Extended Inter-Area ASBR LSA in type 10
    opaque LSA.";

container eia-asbr-opaque {
    container eia-asbr-tlv {
        leaf asbr-rtr-id {
            type rt-types:router-id;
            description
                "The OSPF Router ID of the ASBR.";
        }
    }
    uses faam-sub-tlvs;
    uses ospf:unknown-tlvs;
    description
        "EIA-ASBR TLV, used to advertise additional attributes
        associated with the reachability of an ASBR.";
}

```



```

    uses ospf:unknown-tlvs;

    description
      "OSPFv2 Extended Inter-Area (EIA-ASBR) opaque LSA.";
  }
}

/* FAAM Sub-TLV in OSPFv3 Inter-Area-Router TLV */
augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:areas/ospf:area/ospf:database/"
+ "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
+ "ospf:area-scope-lsa/ospf:version/ospf:ospfv3/"
+ "ospf:ospfv3/ospf:body/ospfv3-e-lsa:e-inter-area-router/"
+ "ospfv3-e-lsa:e-inter-router-tlvs/"
+ "ospfv3-e-lsa:inter-router-tlv" {
when "../../../../../../../../../../../../../../../"
+ "rt:type = 'ospf:ospfv3'" {
  description
    "This augmentation is only valid for OSPFv3
    Inter-Area-Router TLV.";
}
uses faam-sub-tlvs;
description
  "OSPFv3 Area-Scoped Inter-Area-Router TLV.";
}
}
<CODE ENDS>

```

8. 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 [RFC5246].

The NETCONF access control model [RFC6536] 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 the modules 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.

Some of the readable data nodes in the modules 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. This may be undesirable since both due to the fact that exposure may facilitate other attacks. Additionally, network operators may consider their topologies to be sensitive confidential data.

9. IANA Considerations

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

URI: urn:ietf:params:xml:ns:yang:ietf-ospfv3-te
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

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

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

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

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

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

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


```
name: ietf-ospfv3-te
namespace: urn:ietf:params:xml:ns:yang:ietf-ospfv3-te
prefix: ospfv3-te
reference: RFC XXXX

name: ietf-ospf-two-metric
namespace: urn:ietf:params:xml:ns:yang:ietf-ospf-two-metric
prefix: ospf-two-metric
reference: RFC XXXX

name: ietf-ospf-grace-linkdown
namespace: urn:ietf:params:xml:ns:yang:ietf-ospf-grace-linkdown
prefix: ospf-grace-linkdown
reference: RFC XXXX

name: ietf-ospf-lls-localid
namespace: urn:ietf:params:xml:ns:yang:ietf-ospf-lls-localid
prefix: ospf-lls-localid
reference: RFC XXXX

name: ietf-ospf-link-attr
namespace: urn:ietf:params:xml:ns:yang:ietf-ospf-link-attr
prefix: ospf-link-attr
reference: RFC XXXX

name: ietf-ospf-flex-algo
namespace: urn:ietf:params:xml:ns:yang:ietf-ospf-flex-algo
prefix: ospf-flex-algo
reference: RFC XXXX
```

10. Acknowledgements

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

The YANG model was developed using the suite of YANG tools written and maintained by numerous authors.

11. Normative References

[I-D.ietf-lsr-flex-algo]
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and
A. Gulko, "IGP Flexible Algorithm", Work in Progress,
Internet-Draft, draft-ietf-lsr-flex-algo-18, 25 October
2021, <<https://www.ietf.org/archive/id/draft-ietf-lsr-flex-algo-18.txt>>.

- [I-D.ietf-ospf-yang]
Yeung, D., Qu, Y., Zhang, J., Chen, I., and A. Lindem,
"YANG Data Model for OSPF Protocol", Work in Progress,
Internet-Draft, draft-ietf-ospf-yang-29, 17 October 2019,
<<https://www.ietf.org/archive/id/draft-ietf-ospf-yang-29.txt>>.
- [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>>.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security
(TLS) Protocol Version 1.2", RFC 5246,
DOI 10.17487/RFC5246, August 2008,
<<https://www.rfc-editor.org/info/rfc5246>>.
- [RFC5329] Ishiguro, K., Manral, V., Davey, A., and A. Lindem, Ed.,
"Traffic Engineering Extensions to OSPF Version 3",
RFC 5329, DOI 10.17487/RFC5329, September 2008,
<<https://www.rfc-editor.org/info/rfc5329>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for
the Network Configuration Protocol (NETCONF)", RFC 6020,
DOI 10.17487/RFC6020, October 2010,
<<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed.,
and A. Bierman, Ed., "Network Configuration Protocol
(NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011,
<<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure
Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011,
<<https://www.rfc-editor.org/info/rfc6242>>.
- [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>>.

- [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>>.
- [RFC8042] Zhang, Z., Wang, L., and A. Lindem, "OSPF Two-Part Metric", RFC 8042, DOI 10.17487/RFC8042, December 2016, <<https://www.rfc-editor.org/info/rfc8042>>.
- [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>>.
- [RFC8379] Hegde, S., Sarkar, P., Gredler, H., Nanduri, M., and L. Jalil, "OSPF Graceful Link Shutdown", RFC 8379, DOI 10.17487/RFC8379, May 2018, <<https://www.rfc-editor.org/info/rfc8379>>.
- [RFC8510] Psenak, P., Ed., Talaulikar, K., Henderickx, W., and P. Pillay-Esnault, "OSPF Link-Local Signaling (LLS) Extensions for Local Interface ID Advertisement", RFC 8510, DOI 10.17487/RFC8510, January 2019, <<https://www.rfc-editor.org/info/rfc8510>>.
- [RFC8920] Psenak, P., Ed., Ginsberg, L., Henderickx, W., Tantsura, J., and J. Drake, "OSPF Application-Specific Link Attributes", RFC 8920, DOI 10.17487/RFC8920, October 2020, <<https://www.rfc-editor.org/info/rfc8920>>.

Authors' Addresses

Acee Lindem
Cisco Systems
301 Midenhall Way
Cary, NC 27513

Email: acee@cisco.com

Yingzhen Qu
Futurewei
2330 Central Expressway
Santa Clara, CA 95050
United States of America

Email: yingzhen.qu@futurewei.com

Internet
Internet-Draft
Intended status: Standards Track
Expires: 7 September 2022

A. Lindem
Cisco Systems
S. Palani
Microsoft
Y. Qu
Futurewei
6 March 2022

YANG Model for OSPFv3 Extended LSAs
draft-ietf-lsr-ospfv3-extended-lsa-yang-10

Abstract

This document defines a YANG data model augmenting the IETF OSPF YANG model to provide support for OSPFv3 Link State Advertisement (LSA) Extensibility as defined in RFC 8362. OSPFv3 Extended LSAs provide extensible TLV-based LSAs for the base LSA types defined in RFC 5340.

Status of This Memo

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

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

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

This Internet-Draft will expire on 7 September 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Overview	2
1.1. Requirements Language	2
2. Tree Diagrams	3
3. OSPFv3 Extended LSAs	3
4. OSPFv3 Extended LSA Yang Module	8
5. Security Considerations	25
6. IANA Considerations	26
7. Acknowledgements	26
8. References	27
8.1. Normative References	27
8.2. Informative References	28
Authors' Addresses	28

1. Overview

YANG [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 augmenting the IETF OSPF YANG model [I-D.ietf-ospf-yang], which itself augments [RFC8349], to provide support for configuration and operational state for OSPFv3 Extended LSAs as defined in [RFC8362].

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

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] [RFC8174].

2. Tree Diagrams

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

3. OSPFv3 Extended LSAs

This document defines a model for the OSPFv3 Extended LSA feature. It is an augmentation of the OSPF base model provided support for OSPFv3 Link State Advertisement (LSA) Extensibility [RFC8362]. OSPFv3 Extended LSAs provide extensible TLV-based LSAs for the base LSA types defined in [RFC5340].

The OSPFv3 Extended LSA YANG module requires support for the OSPF base model [I-D.ietf-ospf-yang] which defines basic OSPF configuration and state. The OSPF YANG model augments the ietf-routing YANG model defined in [RFC8022]. The augmentations defined in the ietf-ospfv3-extended-lsa YANG model will provide global configuration, area configuration, and addition of OSPFv3 Extended LSAs to the Link State Database (LSDB) operational state.

```

module: ietf-ospfv3-extended-lsa
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/ospf:ospf:
      +---rw extended-lsa-support?  boolean
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area:
      +---rw extended-lsa-support?  boolean
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
    /ospf:interfaces/ospf:interface/ospf:database
    /ospf:link-scope-lsa-type/ospf:link-scope-lsas
    /ospf:link-scope-lsa/ospf:version/ospf:ospfv3
    /ospf:ospfv3/ospf:body:
      +---ro e-link
        +---ro rtr-priority?  uint8
        +---ro lsa-options
        |   +---ro lsa-options*  identityref
        +---ro e-link-tlvs* []
          +---ro unknown-tlv
          |   +---ro type?      uint16
          |   +---ro length?    uint16
          |   +---ro value?     yang:hex-string
          +---ro intra-prefix-tlv
          |   +---ro intra-prefix-tlv-length?  uint16
          |   +---ro metric?                    rt-types:uint24
          |   +---ro prefix?                    inet:ip-prefix
          |   +---ro prefix-options

```



```

    |   +--ro prefix-options*    identityref
    +--ro prefix-length?        uint8
    +--ro sub-tlvs* []
        +--ro unknown-sub-tlv
            +--ro type?          uint16
            +--ro length?        uint16
            +--ro value?         yang:hex-string
    +--ro ipv6-link-local-tlv
        +--ro ipv6-link-local-tlv-length?    uint16
        +--ro link-local-address?             inet:ipv6-address
        +--ro sub-tlvs* []
            +--ro unknown-sub-tlv
                +--ro type?          uint16
                +--ro length?        uint16
                +--ro value?         yang:hex-string
    +--ro ipv4-link-local-tlv
        +--ro ipv4-link-local-tlv-length?    uint16
        +--ro link-local-address?             inet:ipv4-address
        +--ro sub-tlvs* []
            +--ro unknown-sub-tlv
                +--ro type?          uint16
                +--ro length?        uint16
                +--ro value?         yang:hex-string
augment /rt:routing/rt:control-plane-protocols
/rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
/ospf:database/ospf:area-scope-lsa-type
/ospf:area-scope-lsas/ospf:area-scope-lsa
/ospf:version/ospf:ospfv3/ospf:ospfv3/ospf:body:
+--ro e-router
    +--ro router-bits
    |   +--ro rtr-lsa-bits*    identityref
    +--ro lsa-options
    |   +--ro lsa-options*    identityref
    +--ro e-router-tlvs* []
        +--ro unknown-tlv
            +--ro type?          uint16
            +--ro length?        uint16
            +--ro value?         yang:hex-string
        +--ro link-tlv
            +--ro link-tlv-length?    uint16
            +--ro interface-id?        uint32
            +--ro neighbor-interface-id?    uint32
            +--ro neighbor-router-id?    rt-types:router-id
            +--ro type?                  uint8
            +--ro metric?                uint16
            +--ro sub-tlvs* []
                +--ro unknown-sub-tlv
                    +--ro type?          uint16

```



```

|             +---ro length?    uint16
|             +---ro value?     yang:hex-string
+---ro e-network
|   +---ro lsa-options
|   |   +---ro lsa-options*    identityref
+---ro e-network-tlvs* []
|   +---ro unknown--tlv
|   |   +---ro type?          uint16
|   |   +---ro length?       uint16
|   |   +---ro value?        yang:hex-string
+---ro attached-router-tlv
|   +---ro attached-router-tlv-length?    uint16
+---ro Adjacent-neighbor-router-id*    rt-types:router-id
+---ro sub-tlvs* []
|   +---ro unknown-sub-tlv
|   |   +---ro type?          uint16
|   |   +---ro length?       uint16
|   |   +---ro value?        yang:hex-string
+---ro e-inter-area-prefix
|   +---ro e-inter-prefix-tlvs* []
|   +---ro unknown--tlv
|   |   +---ro type?          uint16
|   |   +---ro length?       uint16
|   |   +---ro value?        yang:hex-string
+---ro inter-prefix-tlv
|   +---ro inter-prefix-tlv-length?    uint16
|   +---ro metric?                     rt-types:uint24
|   +---ro prefix?                     inet:ip-prefix
|   +---ro prefix-options
|   |   +---ro prefix-options*    identityref
+---ro prefix-length?                uint8
+---ro sub-tlvs* []
|   +---ro unknown-sub-tlv
|   |   +---ro type?          uint16
|   |   +---ro length?       uint16
|   |   +---ro value?        yang:hex-string
+---ro e-inter-area-router
|   +---ro e-inter-router-tlvs* []
|   +---ro unknown-tlv
|   |   +---ro type?          uint16
|   |   +---ro length?       uint16
|   |   +---ro value?        yang:hex-string
+---ro inter-router-tlv
|   +---ro inter-router-tlv-length?    uint16
|   +---ro router-bits
|   |   +---ro rtr-lsa-bits*    identityref
+---ro lsa-options
|   +---ro lsa-options*    identityref

```



```

    +--ro metric?                               rt-types:uint24
    +--ro destination-router-id?                 rt-types:router-id
    +--ro sub-tlvs* []
      +--ro unknown-sub-tlv
        +--ro type?          uint16
        +--ro length?        uint16
        +--ro value?         yang:hex-string
+--ro e-intra-area-prefix
  +--ro referenced-ls-type?          uint16
  +--ro referenced-link-state-id?    uint32
  +--ro referenced-adv-router?       rt-types:router-id
  +--ro e-intra-prefix-tlvs* []
    +--ro unknown-tlv
      +--ro type?          uint16
      +--ro length?        uint16
      +--ro value?         yang:hex-string
    +--ro intra-prefix-tlv
      +--ro intra-prefix-tlv-length?  uint16
      +--ro metric?                  rt-types:uint24
      +--ro prefix?                  inet:ip-prefix
      +--ro prefix-options
        | +--ro prefix-options*      identityref
      +--ro prefix-length?            uint8
      +--ro sub-tlvs* []
        +--ro unknown-sub-tlv
          +--ro type?          uint16
          +--ro length?        uint16
          +--ro value?         yang:hex-string
augment /rt:routing/rt:control-plane-protocols
/rt:control-plane-protocol/ospf:ospf/ospf:database
/ospf:as-scope-lsa-type/ospf:as-scope-lsas
/ospf:as-scope-lsa/ospf:version/ospf:ospfv3
/ospf:ospfv3/ospf:body:
+--ro e-as-external
  +--ro e-external-tlvs* []
    +--ro unknown-tlv
      +--ro type?          uint16
      +--ro length?        uint16
      +--ro value?         yang:hex-string
    +--ro external-prefix-tlv
      +--ro external-prefix-tlv-length?  uint16
      +--ro flags
        | +--ro ospfv3-e-external-prefix-bits*  identityref
      +--ro metric?                  rt-types:uint24
      +--ro prefix?                  inet:ip-prefix
      +--ro prefix-options
        | +--ro prefix-options*      identityref
      +--ro prefix-length?            uint8

```



```

    +--ro sub-tlvs* []
      +--ro unknown-sub-tlv
        | +--ro type?      uint16
        | +--ro length?    uint16
        | +--ro value?     yang:hex-string
      +--ro ipv6-fwd-addr-sub-tlv
        | +--ro ipv6-fwd-addr-sub-tlv-length?  uint16
        | +--ro forwarding-address?            inet:ipv6-address
      +--ro ipv4-fwd-addr-sub-tlv
        | +--ro ipv4-fwd-addr-sub-tlv-length?  uint16
        | +--ro forwarding-address?            inet:ipv4-address
      +--ro route-tag-sub-tlv
        | +--ro route-tag-sub-tlv-length?      uint16
        | +--ro route-tag?                      uint32
+--ro e-nssa
  +--ro e-external-tlvs* []
    +--ro unknown-tlv
      | +--ro type?      uint16
      | +--ro length?    uint16
      | +--ro value?     yang:hex-string
    +--ro external-prefix-tlv
      +--ro external-prefix-tlv-length?  uint16
      +--ro flags
      | +--ro ospfv3-e-external-prefix-bits*  identityref
      +--ro metric?                          rt-types:uint24
      +--ro prefix?                          inet:ip-prefix
      +--ro prefix-options
      | +--ro prefix-options*  identityref
      +--ro prefix-length?      uint8
      +--ro sub-tlvs* []
        +--ro unknown-sub-tlv
          | +--ro type?      uint16
          | +--ro length?    uint16
          | +--ro value?     yang:hex-string
        +--ro ipv6-fwd-addr-sub-tlv
          | +--ro ipv6-fwd-addr-sub-tlv-length?  uint16
          | +--ro forwarding-address?            inet:ipv6-address
        +--ro ipv4-fwd-addr-sub-tlv
          | +--ro ipv4-fwd-addr-sub-tlv-length?  uint16
          | +--ro forwarding-address?            inet:ipv4-address
        +--ro route-tag-sub-tlv
          +--ro route-tag-sub-tlv-length?      uint16
          +--ro route-tag?                      uint32

```


4. OSPFv3 Extended LSA Yang Module

The following RFCs and drafts are not referenced in the document text but are referenced in the `ietf-ospfv3-extended-lsa.yang` module: [RFC6991], [RFC8294].

```
<CODE BEGINS> file "ietf-ospfv3-extended-lsa@2022-03-06.yang"
module ietf-ospfv3-extended-lsa {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-ospfv3-extended-lsa";

  prefix ospfv3-e-lsa;

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

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

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

  import ietf-ospf {
    prefix "ospf";
    reference "RFC YYYY - A YANG Data Model for OSPF
              Protocol";
  }

  organization
    "IETF LSR - Link State Routing Working Group";

  contact
    "WG Web:    <http://tools.ietf.org/wg/lsr/>
    WG List:    <mailto:lsr@ietf.org>

    Author:     Acee Lindem
                <mailto:acee@cisco.com>
    Author:     Sharmila Palani
                <mailto:sharmila.palani@microsoft.com>
```


Author: Yingzhen Qu
<mailto:yingzhen.qu@futurewei.com>;

description

"This YANG module defines the configuration and operational state for OSPFv3 Extended LSAs, which is common across all of the vendor implementations. The semantics and encodings for OSPFv3 Extended LSAs is described in RFC 8362.

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

Copyright (c) 2022 IETF Trust and the persons identified as authors of the code. All rights reserved.

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

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

reference "RFC XXXX - YANG Model for OSPFv3 Extended LSAs";

revision 2022-03-06 {

description

"Initial revision.";

reference

"RFC XXXX: A YANG Data Model for OSPFv3 Extended LSAs.";

}

/*

* OSPFv3 Extend LSA Type Identities

*/

identity ospfv3-e-router-lsa {

base ospf:ospfv3-lsa-type;

description

"OSPFv3 Extended Router LSA - Type 0xA021";

reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA) Extensibility, Section 4.1";

}

identity ospfv3-e-network-lsa {

base ospf:ospfv3-lsa-type;


```
    description
      "OSPFv3 Extended Network LSA - Type 0xA022";
    reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
      Extensibility, Section 4.2";
  }

  identity ospfv3-e-summary-lsa-type {
    base ospf:ospfv3-lsa-type;
    description
      "OSPFv3 Extended Summary LSA types";
    reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
      Extensibility, Section 4.3 and Section 4.4";
  }

  identity ospfv3-e-inter-area-prefix-lsa {
    base ospfv3-e-summary-lsa-type;
    description
      "OSPFv3 Extended Inter-area Prefix LSA - Type 0xA023";
    reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
      Extensibility, Section 4.3";
  }

  identity ospfv3-e-inter-area-router-lsa {
    base ospfv3-e-summary-lsa-type;
    description
      "OSPFv3 Extended Inter-area Router LSA - Type 0xA024";
    reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
      Extensibility, Section 4.3";
  }

  identity ospfv3-e-external-lsa-type {
    base ospf:ospfv3-lsa-type;
    description
      "OSPFv3 Extended External LSA types";
    reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
      Extensibility, Section 4.5 and Section 4.6";
  }

  identity ospfv3-e-as-external-lsa {
    base ospfv3-e-external-lsa-type;
    description
      "OSPFv3 Extended AS-External LSA - Type 0xC025";
    reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
      Extensibility, Section 4.5";
  }

  identity ospfv3-e-nssa-lsa {
    base ospfv3-e-external-lsa-type;
```



```
    description
      "OSPFv3 Extended Not-So-Stubby-Area (NSSA) LSA -
       Type 0xA027";
    reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
      Extensibility, Section 4.6";
  }

  identity ospfv3-e-link-lsa {
    base ospf:ospfv3-lsa-type;
    description
      "OSPFv3 Extended Link LSA - Type 0x8028";
    reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
      Extensibility, Section 4.7";
  }

  identity ospfv3-e-intra-area-prefix-lsa {
    base ospf:ospfv3-lsa-type;
    description
      "OSPFv3 Extended Intra-area Prefix LSA - Type 0xA029";
    reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
      Extensibility, Section 4.8";
  }

  identity ospfv3-e-prefix-option {
    description
      "Base identity for OSPFv3 Prefix Options.";
    reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
      Extensibility, Section 3.1";
  }

  identity nu-bit {
    base ospfv3-e-prefix-option;
    description
      "When set, the prefix should be excluded
       from IPv6 unicast calculations.";
    reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
      Extensibility, Section 3.1";
  }

  identity la-bit {
    base ospfv3-e-prefix-option;
    description
      "When set, the prefix is actually an IPv6 interface
       address of the Advertising Router.";
    reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
      Extensibility, Section 3.1";
  }
}
```



```
identity p-bit {
  base ospfv3-e-prefix-option;
  description
    "When set, the NSSA area prefix should be
     translated to an AS External LSA and advertised
     by the translating NSSA Border Router.";
  reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
            Extensibility, Section 3.1";
}

identity dn-bit {
  base ospfv3-e-prefix-option;
  description
    "When set, the inter-area-prefix LSA or
     AS-external LSA prefix has been advertised as an
     L3VPN prefix.";
  reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
            Extensibility, Section 3.1";
}

identity n-bit {
  base ospfv3-e-prefix-option;
  description
    "When set, the prefix is a host address that identifies
     the advertising router.";
  reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
            Extensibility, Section 3.1";
}

identity ospfv3-e-external-prefix-option {
  description
    "Base identity for OSPFv3 External Prefix Options.";
  reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
            Extensibility, Section 3.6";
}

identity e-bit {
  base ospfv3-e-external-prefix-option;
  description
    "When set, the metric specified is a Type 2
     external metric.";
  reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
            Extensibility, Section 3.6";
}

grouping unknown-sub-tlv {
  description
    "Unknown TLV grouping";
}
```



```
    container unknown-sub-tlv {
      uses ospf:tlv;
      description "Unknown External TLV Sub-TLV";
    }
    reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
              Extensibility, Section 6.3";
  }

  grouping ospfv3-lsa-prefix {
    description
      "OSPFv3 LSA prefix";

    leaf prefix {
      type inet:ip-prefix;
      description
        "LSA Prefix";
    }
    container prefix-options {
      leaf-list prefix-options {
        type identityref {
          base ospfv3-e-prefix-option;
        }
        description
          "OSPFv3 prefix option flag list. This list will
           contain the identities for the OSPFv3 options
           that are set for the OSPFv3 prefix.";
      }
      description "Prefix options.";
      reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
                Extensibility, Section 3.1";
    }

    leaf prefix-length {
      type uint8 {
        range "0..128";
      }
      description "Prefix length.";
    }
    reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
              Extensibility, Section 3";
  }

  grouping ipv6-fwd-addr-sub-tlv {
    container ipv6-fwd-addr-sub-tlv {
      description
        "IPv6 Forwarding Address Sub-TLV";
      leaf ipv6-fwd-addr-sub-tlv-length {
        type uint16;
      }
    }
  }
```



```
        description
            "IPv6 Forwarding Addrss Sub-TLV Length - 16
             for IPv6 address";
    }
    leaf forwarding-address {
        type inet:ipv6-address;
        description
            "Forwarding address";
    }
}
description
    "IPv6 Forwarding Address Sub-TLV grouping";
reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
          Extensibility, Section 3.10";
}

grouping ipv4-fwd-addr-sub-tlv {
    container ipv4-fwd-addr-sub-tlv {
        description
            "IPv4 Forwarding Address Sub-TLV";
        leaf ipv4-fwd-addr-sub-tlv-length {
            type uint16;
            description
                "IPv4 Forwarding Addrss Sub-TLV Length - 4
                 for IPv4 address";
        }
        leaf forwarding-address {
            type inet:ipv4-address;
            description
                "Forwarding address";
        }
    }
}
description
    "IPv4 Forwarding Address Sub-TLV grouping";
reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
          Extensibility, Section 3.11";
}

grouping route-tag-sub-tlv {
    container route-tag-sub-tlv {
        description
            "Route Tag Sub-TLV";
        leaf route-tag-sub-tlv-length {
            type uint16;
            description
                "Route Tag Sub-TLV Length - 4 for 32-bit tag";
        }
        leaf route-tag {
```



```
        type uint32;
        description
            "Route Tag";
    }
}
description
    "Route Tag Sub-TLV grouping";
reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
          Extensibility, Section 3.12";
}

grouping external-prefix-tlv {
    container external-prefix-tlv {
        description "External Prefix LSA TLV";
        leaf external-prefix-tlv-length {
            type uint16;
            description
                "External Prefix TLV Length - Variable dependent
                 on sub-TLVs";
        }
        container flags {
            leaf-list ospfv3-e-external-prefix-bits {
                type identityref {
                    base ospfv3-e-external-prefix-option;
                }
                description "OSPFv3 external-prefix TLV bits list.";
            }
            description "External Prefix Flags";
        }
        leaf metric {
            type rt-types:uint24;
            description "External Prefix Metric";
        }
    }
    uses ospfv3-lsa-prefix;
    list sub-tlvs {
        description "External Prefix TLV Sub-TLVs";
        uses unknown-sub-tlv;
        uses ipv6-fwd-addr-sub-tlv;
        uses ipv4-fwd-addr-sub-tlv;
        uses route-tag-sub-tlv;
    }
}
description "External Prefix TLV Grouping";
reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
          Extensibility, Section 3.6";
}

grouping intra-area-prefix-tlv {
```



```
container intra-prefix-tlv {
  description "Intra-Area Prefix LSA TLV";
  leaf intra-prefix-tlv-length {
    type uint16;
    description
      "Intra-Area Prefix TLV Length - Variable dependent
       on sub-TLVs";
  }
  leaf metric {
    type rt-types:uint24;
    description "Intra-Area Prefix Metric";
  }
  uses ospfv3-lsa-prefix;
  list sub-tlvs {
    description "Intra-Area Prefix TLV Sub-TLVs";
    uses unknown-sub-tlv;
  }
}
description "Intra-Area Prefix TLV Grouping";
reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
  Extensibility, Section 3.4";
}

grouping ipv6-link-local-tlv {
  container ipv6-link-local-tlv {
    description "IPv6 Link-Local LSA TLV";
    leaf ipv6-link-local-tlv-length {
      type uint16;
      description
        "IPv6 Link-Local TLV Length - Variable dependent
         on sub-TLVs";
    }
    leaf link-local-address {
      type inet:ipv6-address;
      description
        "IPv6 Link Local address";
    }
    list sub-tlvs {
      description "IPv6 Link Local TLV Sub-TLVs";
      uses unknown-sub-tlv;
    }
  }
  description "IPv6 Link-Local TLV Grouping";
  reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
    Extensibility, Section 3.8";
}

grouping ipv4-link-local-tlv {
```



```
container ipv4-link-local-tlv {
  description "IPv4 Link-Local LSA TLV";
  leaf ipv4-link-local-tlv-length {
    type uint16;
    description
      "IPv4 Link-Local TLV Length - Variable dependent
       on sub-TLVs";
  }
  leaf link-local-address {
    type inet:ipv4-address;
    description
      "IPv4 Link Local address";
  }
  list sub-tlvs {
    description "IPv4 Link Local TLV Sub-TLVs";
    uses unknown-sub-tlv;
  }
}
description "IPv4 Link-Local TLV Grouping";
reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
  Extensibility, Section 3.9";
}

grouping ospfv3-e-lsa-area {
  description "Area scope OSPFv3 Extended LSAs.";
  reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
    Extensibility, Section 4";
}

container e-router {
  when "../../ospf:header/ospf:type = "
    + "'ospfv3-e-lsa:ospfv3-e-router-lsa'" {
    description "Only valid for OSPFv3 Extended-Router LSAs";
  }
  description "OSPFv3 Extended Router LSA";
  reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
    Extensibility, Section 4.1";
  uses ospf:ospf-router-lsa-bits;
  uses ospf:ospfv3-lsa-options;

  list e-router-tlvs {
    description "E-Router LSA TLVs";
    container unknown-tlv {
      uses ospf:tlv;
      description "Unknown E-Router TLV";
    }
    container link-tlv {
      description "E-Router LSA TLV";
      leaf link-tlv-length {
```



```

        type uint16;
        description
            "Link TLV Length - Variable dependent on sub-TLVs";
    }
    leaf interface-id {
        type uint32;
        description "Interface ID for link";
    }
    leaf neighbor-interface-id {
        type uint32;
        description "Neighbor's Interface ID for link";
    }
    leaf neighbor-router-id {
        type rt-types:router-id;
        description "Neighbor's Router ID for link";
    }
    leaf type {
        type ospf:router-link-type;
        description "Link type: 1 - Point-to-Point Link
                    2 - Transit Network Link
                    3 - Stub Network Link Link
                    4 - Virtual Link";
    }
    leaf metric {
        type uint16;
        description "Link Metric";
    }
    list sub-tlvs {
        description "Link TLV Sub-TLVs";
        uses unknown-sub-tlv;
    }
}

container e-network {
    when "../../ospf:header/ospf:type = "
        + "'ospfv3-e-lsa:ospfv3-e-network-lsa'" {
        description
            "Only applies to E-Network LSAs.";
    }
    description "Extended Network LSA";
    reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
        Extensibility, Section 4.2";
    uses ospf:ospfv3-lsa-options;
    list e-network-tlvs {
        description "E-Network LSA TLVs";
        container unknown--tlv {

```



```
        uses ospf:tlv;
        description "Unknown E-Network TLV";
    }
    container attached-router-tlv {
        description "Attached Router TLV";
        leaf attached-router-tlv-length {
            type uint16;
            description
                "Attached Router TLV Length - Variable dependent
                on sub-TLVs";
        }
        leaf-list Adjacent-neighbor-router-id {
            type rt-types:router-id;
            description "Adjacent Neighbor's Router ID";
        }
        list sub-tlvs {
            description "Attached Router TLV Sub-TLVs";
            uses unknown-sub-tlv;
        }
    }
}

container e-inter-area-prefix {
    when "../../ospf:header/ospf:type = "
        + "'ospfv3-e-lsa:ospfv3-e-inter-area-prefix-lsa'" {
        description
            "Only applies to E-Inter-Area-Prefix LSAs.";
    }
    description "Extended Inter-Area Prefix LSA";
    reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
        Extensibility, Section 4.3";
    list e-inter-prefix-tlvs {
        description "E-Inter-Area-Prefix LSA TLVs";
        container unknown--tlv {
            uses ospf:tlv;
            description "Unknown E-Inter-Area-Prefix TLV";
        }
        container inter-prefix-tlv {
            description "Unknown E-Inter-Area-Prefix LSA TLV";
            leaf inter-prefix-tlv-length {
                type uint16;
                description
                    "Inter-Area-Prefix TLV Length - Variable dependent
                    on sub-TLVs";
            }
        }
        leaf metric {
            type rt-types:uint24;
        }
    }
}
```



```
        description "Inter-Area Prefix Metric";
    }
    uses ospfv3-lsa-prefix;
    list sub-tlvs {
        description "Inter-Area Prefix TLV Sub-TLVs";
        uses unknown-sub-tlv;
    }
}
}

container e-inter-area-router {
    when "../../ospf:header/ospf:type = "
        + "'ospfv3-e-lsa:ospfv3-e-inter-area-router-lsa'" {
        description
            "Only applies to E-Inter-Area-Router LSAs.";
    }
    description "Extended Inter-Area Router LSA";
    reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
        Extensibility, Section 4.4";
    list e-inter-router-tlvs {
        description "E-Inter-Area-Router LSA TLVs";
        container unknown-tlv {
            uses ospf:tlv;
            description "Unknown E-Inter-Area-Router TLV";
        }
        container inter-router-tlv {
            description "Unknown E-Inter-Area-Router LSA TLV";
            leaf inter-router-tlv-length {
                type uint16;
                description
                    "Inter-Area-Router TLV Length - Variable dependent
                    on sub-TLVs";
            }
            uses ospf:ospf-router-lsa-bits;
            uses ospf:ospfv3-lsa-options;
            leaf metric {
                type rt-types:uint24;
                description "Inter-Area Router Metric";
            }
            leaf destination-router-id {
                type rt-types:router-id;
                description "Destination Router ID";
            }
            list sub-tlvs {
                description "Inter-Area Router TLV Sub-TLVs";
                uses unknown-sub-tlv;
            }
        }
    }
}
```



```
    }
  }
}

container e-intra-area-prefix {
  when "../../ospf:header/ospf:type = "
    + "'ospfv3-e-lsa:ospfv3-e-intra-area-prefix-lsa'" {
    description
      "Only applies to E-Intra-Area-Prefix LSAs.";
  }
  description "E-Intra-Area-Prefix LSA";
  reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
    Extensibility, Section 4.8";
  leaf referenced-ls-type {
    type uint16;
    description "Referenced Link State type";
  }
  leaf referenced-link-state-id {
    type uint32;
    description
      "Referenced Link State ID";
  }
  leaf referenced-adv-router {
    type rt-types:router-id;
    description
      "Referenced Advertising Router";
  }
  list e-intra-prefix-tlvs {
    description "E-Intra-Area-Prefix LSA TLVs";
    container unknown-tlv {
      uses ospf:tlv;
      description "Unknown E-Intra-Area-Prefix TLV";
    }
    uses intra-area-prefix-tlv;
  }
}

grouping ospfv3-e-lsa-as {
  description "AS scope OSPFv3 Extended LSAs.";
  container e-as-external {
    when "../../ospf:header/ospf:type = "
      + "'ospfv3-e-lsa:ospfv3-e-as-external-lsa'" {
      description
        "Only applies to E-AS-external LSAs.";
    }
    list e-external-tlvs {
      description "E-External LSA TLVs";
    }
  }
}
```



```
        container unknown-tlv {
            uses ospf:tlv;
            description "Unknown E-External TLV";
        }
        uses external-prefix-tlv;
    }
    description "E-AS-External LSA.";
    reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
        Extensibility, Section 4.5";
}

container e-nssa {
    when "../../ospf:header/ospf:type = "
        + "'ospfv3-e-lsa:ospfv3-e-nssa-lsa'" {
        description
            "Only applies to E-NSSA LSAs.";
    }
    list e-external-tlvs {
        description "E-NSSA LSA TLVs";
        container unknown-tlv {
            uses ospf:tlv;
            description "Unknown E-External TLV";
        }
        uses external-prefix-tlv;
    }
    description "E-NSSA LSA.";
    reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
        Extensibility, Section 4.6";
}

}

grouping ospfv3-e-lsa-link {
    description "Link scope OSPFv3 Extended LSAs.";
    container e-link {
        when "../../ospf:header/ospf:type = "
            + "'ospfv3-e-lsa:ospfv3-e-link-lsa'" {
            description
                "Only applies to Extended Link LSAs.";
        }
        description "E-Link LSA";
        reference "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
            Extensibility, Section 4.7";
        leaf rtr-priority {
            type uint8;
            description "Router Priority for the interface.";
        }
        uses ospf:ospfv3-lsa-options;
        list e-link-tlvs {
```



```
        description "E-Link LSA TLVs";
        container unknown-tlv {
            uses ospf:tlv;
            description "Unknown E-Link TLV";
        }
        uses intra-area-prefix-tlv;
        uses ipv6-link-local-tlv;
        uses ipv4-link-local-tlv;
    }
}

/* Configuration */
augment "/rt:routing/rt:control-plane-protocols"
    + "/rt:control-plane-protocol/ospf:ospf" {
    when "../rt:type = 'ospf:ospfv3'" {
        description
            "This augments the OSPFv3 routing protocol when used.";
    }
    description
        "This augments the OSPFv3 protocol configuration
        with extended LSA support.";
    leaf extended-lsa-support {
        type boolean;
        default false;
        description
            "Enable OSPFv3 Extended LSA Support for the OSPFv3
            domain";
        reference
            "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
            Extensibility, Appendix B - ExtendedLSASupport";
    }
}

augment "/rt:routing/rt:control-plane-protocols/"
    + "rt:control-plane-protocol/ospf:ospf:areas/ospf:area" {
    when "../../../../rt:type = 'ospf:ospfv3'" {
        description
            "This augments the OSPFv3 area configuration
            when used.";
    }
    description
        "This augments the OSPFv3 protocol area
        configuration with extended LSA support.";
    leaf extended-lsa-support {
        type boolean;
        must "derived-from(..ospf:area-type,'stub-nssa-area') or "
            + "(current() = 'true') or "
```



```

    + "(../../../../../extended-lsa-support = 'false')" {
description
    "For regular areas, i.e., areas where AS-scoped LSAs
    disabling AreaExtendedLSASupport for a regular
    OSPFv3 area (not a Stub or NSSA area) when
    ExtendedLSASupport is enabled is contradictory and
    is prohibited.";
    }
description
    "Enable OSPFv3 Extended LSA Support for the OSPFv3
    area. If not specified, Extended LSA support status
    is inherited from the instance level configuration.";
reference
    "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
    Extensibility, Appendix B - AreaExtendedLSASupport";
    }
}

/*
 * Link State Database (LSDB) Augmentations
 */
augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:areas/ospf:area/"
+ "ospf:interfaces/ospf:interface/ospf:database/"
+ "ospf:link-scope-lsa-type/ospf:link-scope-lsas/"
+ "ospf:link-scope-lsa/ospf:version/ospf:ospfv3/"
+ "ospf:ospfv3/ospf:body" {
when "../../../../../../../../../../../../../"
+ "rt:type = 'ospf:ospfv3'" {
description
    "This augmentation is only valid for OSPFv3.";
}
description
    "OSPFv3 Link-Scoped Extended LSAs";

uses ospfv3-e-lsa-link;
}

augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:areas/ospf:area/ospf:database/"
+ "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
+ "ospf:area-scope-lsa/ospf:version/ospf:ospfv3/"
+ "ospf:ospfv3/ospf:body" {
when "../../../../../../../../../../../../../"
+ "rt:type = 'ospf:ospfv3'" {
description

```



```
        "This augmentation is only valid for OSPFv3.";
    }
    description
        "OSPFv3 Area-Scoped Extended LSAs";

    uses ospfv3-e-lsa-area;
}

augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:database/"
+ "ospf:as-scope-lsa-type/ospf:as-scope-lsas/"
+ "ospf:as-scope-lsa/ospf:version/ospf:ospfv3/"
+ "ospf:ospfv3/ospf:body" {
    when "../.../.../.../.../.../..."
    + "rt:type = 'ospf:ospfv3'" {
        description
            "This augmentation is only valid for OSPFv3.";
    }
    description
        "OSPFv3 AS-Scoped Extended LSAs";

    uses ospfv3-e-lsa-as;
}
}
<CODE ENDS>
```

5. 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 [RFC6536] 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-ospfv3-extended-lsa.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. There are the subtrees and data nodes and their sensitivity/vulnerability:

```
/ospf:ospf/extended-lsa-support
```

```
/ospf:ospf/ospf:areas/ospf:area/extended-lsa-support - For OSPFv3  
Extended LSAs, the ability to disable OSPFv3 Extended LSA support  
result in a denial of service.
```

Some of the readable data nodes in the `ietf-ospfv3-extended-lsa.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 and information beyond the scope of OSPF router. This may be undesirable since both due to the fact that exposure may facilitate other attacks. Additionally, network operators may consider their topologies to be sensitive confidential data.

6. IANA Considerations

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

```
URI: urn:ietf:params:xml:ns:yang:ietf-ospfv3-extended-lsa  
Registrant Contact: The IESG.  
XML: N/A, the requested URI is an XML namespace.
```

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

```
name: ietf-ospfv3-extended-lsa  
namespace: urn:ietf:params:xml:ns:yang:ietf-ospfv3-extended-lsa  
prefix: ospfv3-e-lsa  
reference: RFC XXXX
```

7. Acknowledgements

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

The YANG model was developed using the suite of YANG tools written and maintained by numerous authors.

Thanks much to Tom Petch for his review and comments.

8. References

8.1. Normative References

- [I-D.ietf-ospf-yang]
Yeung, D., Qu, Y., Zhang, J., Chen, I., and A. Lindem,
"YANG Data Model for OSPF Protocol", Work in Progress,
Internet-Draft, draft-ietf-ospf-yang-29, 17 October 2019,
<<https://www.ietf.org/archive/id/draft-ietf-ospf-yang-29.txt>>.
- [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>>.
- [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF
for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008,
<<https://www.rfc-editor.org/info/rfc5340>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for
the Network Configuration Protocol (NETCONF)", RFC 6020,
DOI 10.17487/RFC6020, October 2010,
<<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed.,
and A. Bierman, Ed., "Network Configuration Protocol
(NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011,
<<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure
Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011,
<<https://www.rfc-editor.org/info/rfc6242>>.
- [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>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types",
RFC 6991, DOI 10.17487/RFC6991, July 2013,
<<https://www.rfc-editor.org/info/rfc6991>>.

- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8022] Lhotka, L. and A. Lindem, "A YANG Data Model for Routing Management", RFC 8022, DOI 10.17487/RFC8022, November 2016, <<https://www.rfc-editor.org/info/rfc8022>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [RFC8362] Lindem, A., Roy, A., Goethals, D., Reddy Vallem, V., and F. Baker, "OSPFv3 Link State Advertisement (LSA) Extensibility", RFC 8362, DOI 10.17487/RFC8362, April 2018, <<https://www.rfc-editor.org/info/rfc8362>>.
- [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>>.

8.2. Informative References

- [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>>.

Authors' Addresses

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

Sharmila Palani
Microsoft
1 Microsoft Way
Redmond, WA 98052
Email: sharmila.palani@microsoft.com

Yingzhen Qu
Futurewei
2330 Central Expressway
Santa Clara, CA 95050
United States of America
Email: yingzhen.qu@futurewei.com

Internet
Internet-Draft
Intended status: Standards Track
Expires: 6 July 2022

D. Yeung
Arrcus
Y. Qu
Futurewei
J. Zhang
Juniper Networks
I. Chen
The MITRE Corporation
A. Lindem
Cisco Systems
2 January 2022

YANG Data Model for OSPF Segment Routing
draft-ietf-ospf-sr-yang-17

Abstract

This document defines a YANG data module that can be used to configure and manage OSPF Extensions for Segment Routing. It also defines a module for management of Signaling Maximum SID Depth (MSD) Using OSPF.

Status of This Memo

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

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

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

This Internet-Draft will expire on 6 July 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document.

Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Overview	2
1.1. Requirements Language	3
1.2. Tree Diagrams	3
2. OSPF MSD	3
2.1. OSPF MSD YANG Module	4
3. OSPF Segment Routing	11
3.1. OSPF Segment Routing YANG Module	16
4. Security Considerations	30
5. Acknowledgements	31
6. IANA Considerations	31
7. References	32
7.1. Normative References	32
7.2. Informative References	34
Appendix A. Contributors' Addresses	34
Authors' Addresses	34

1. Overview

YANG [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 OSPFv2 extensions for Segment Routing [RFC8665] and it is an augmentation to the OSPF YANG data model.

This document also defines a YANG data model for Signaling Maximum SID Depth (MSD) Using OSPF [RFC8476], which augments the base OSPF YANG data model.

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

1.1. Requirements Language

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

1.2. Tree Diagrams

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

2. OSPF MSD

This document defines a model for Signaling Maximum SID Depth (MSD) Using OSPF [RFC8476]. It is an augmentation of the OSPF base model.

```

module: ietf-ospf-msd
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
    /ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
    /ospf:area-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
    /ospf:body/ospf:opaque/ospf:ri-opaque:
  +--ro node-msd-tlv
    +--ro node-msds* [msd-type]
      +--ro msd-type      identityref
      +--ro msd-value?    uint8
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/ospf:ospf/ospf:database
    /ospf:as-scope-lsa-type/ospf:as-scope-lsas/ospf:as-scope-lsa
    /ospf:version/ospf:ospfv2/ospf:ospfv2/ospf:body/ospf:opaque
    /ospf:ri-opaque:
  +--ro node-msd-tlv
    +--ro node-msds* [msd-type]
      +--ro msd-type      identityref
      +--ro msd-value?    uint8
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
    /ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
    /ospf:area-scope-lsa/ospf:version/ospf:ospfv3/ospf:ospfv3
    /ospf:body/ospf:router-information:
  +--ro node-msd-tlv
    +--ro node-msds* [msd-type]
      +--ro msd-type      identityref
      +--ro msd-value?    uint8
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/ospf:ospf/ospf:database
    /ospf:as-scope-lsa-type/ospf:as-scope-lsas/ospf:as-scope-lsa
    /ospf:version/ospf:ospfv3/ospf:ospfv3/ospf:body

```



```

        /ospf:router-information:
+--ro node-msd-tlv
  +--ro node-msds* [msd-type]
    +--ro msd-type      identityref
    +--ro msd-value?    uint8
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
  /ospf:interfaces/ospf:interface/ospf:database
  /ospf:link-scope-lsa-type/ospf:link-scope-lsas
  /ospf:link-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
  /ospf:body/ospf:opaque/ospf:extended-link-opaque
  /ospf:extended-link-tlv:
+--ro link-msd-sub-tlv
  +--ro link-msds* [msd-type]
    +--ro msd-type      identityref
    +--ro msd-value?    uint8
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
  /ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
  /ospf:area-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
  /ospf:body/ospf:opaque/ospf:extended-link-opaque
  /ospf:extended-link-tlv:
+--ro link-msd-sub-tlv
  +--ro link-msds* [msd-type]
    +--ro msd-type      identityref
    +--ro msd-value?    uint8
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/ospf:ospf/ospf:database
  /ospf:as-scope-lsa-type/ospf:as-scope-lsas/ospf:as-scope-lsa
  /ospf:version/ospf:ospfv2/ospf:ospfv2/ospf:body/ospf:opaque
  /ospf:extended-link-opaque/ospf:extended-link-tlv:
+--ro link-msd-sub-tlv
  +--ro link-msds* [msd-type]
    +--ro msd-type      identityref
    +--ro msd-value?    uint8
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
  /ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
  /ospf:area-scope-lsa/ospf:version/ospf:ospfv3/ospf:ospfv3
  /ospf:body/ospfv3-e-lsa:e-router/ospfv3-e-lsa:e-router-tlvs:
+--ro link-msd-sub-tlv
  +--ro link-msds* [msd-type]
    +--ro msd-type      identityref
    +--ro msd-value?    uint8

```

2.1. OSPF MSD YANG Module


```
<CODE BEGINS> file "ietf-ospf-msd@2022-01-02.yang"
module ietf-ospf-msd {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-ospf-msd";
  prefix ospf-msd;

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

  import ietf-ospf {
    prefix ospf;
  }

  import ietf-ospfv3-extended-lsa {
    prefix ospfv3-e-lsa;
  }

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

    Author:   Yingzhen Qu
              <mailto:yingzhen.qu@futurewei.com>
    Author:   Acee Lindem
              <mailto:acee@cisco.com>
    Author:   Stephane Litkowski
              <mailto:slitkows.ietf@gmail.com>
    Author:   Jeff Tantsura
              <jefftant.ietf@gmail.com>

";
  description
    "The YANG module augments the base OSPF model to
    manage different types of MSDs.

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

    Copyright (c) 2022 IETF Trust and the persons identified as
    authors of the code. All rights reserved.

    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject to
```


the license terms contained in, the Revised BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; 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.";

```
reference "RFC XXXX: YANG Data Model for OSPF MSD.";

revision 2022-01-02 {
  description
    "Initial Version";
  reference "RFC XXXX: YANG Data Model for OSPF MSD.";
}

identity msd-base-type {
  description
    "Base identity for MSD Type";
}

identity base-mpls-msd {
  base msd-base-type;
  description
    "Base MPLS Imposition MSD.";
  reference
    "RFC 8491: Singling MSD using IS-IS.";
}

identity erld-msd {
  base msd-base-type;
  description
    "ERLD-MSD is defined to advertise the ERLD.";
  reference
    "RFC 8662: Entropy Label for Source Packet Routing in
      Networking (SPRING) Tunnels";
}

grouping node-msd-tlv {
  description
    "Grouping for node MSD.";
```



```
    container node-msd-tlv {
      list node-msds {
        key "msd-type";
        leaf msd-type {
          type identityref {
            base msd-base-type;
          }
          description
            "MSD-Types";
        }
        leaf msd-value {
          type uint8;
          description
            "MSD value, in the range of 0-255.";
        }
        description
          "Node MSD is the smallest link MSD supported by
           the node.";
      }
      description
        "Node MSD is the number of SIDs supported by a node.";
      reference
        "RFC 8476: Signaling Maximum SID Depth (MSD) Using OSPF";
    }
  }

  grouping link-msd-sub-tlv {
    description
      "Link Maximum SID Depth (MSD) grouping for an interface.";
    container link-msd-sub-tlv {
      list link-msds {
        key "msd-type";
        leaf msd-type {
          type identityref {
            base msd-base-type;
          }
          description
            "MSD-Types";
        }
        leaf msd-value {
          type uint8;
          description
            "MSD value, in the range of 0-255.";
        }
        description
          "List of link MSDs";
      }
    }
    description
```



```

        "Link MSD sub-tlvs.";
    }
}

/* Node MSD TLV */
augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:areas/"
+ "ospf:area/ospf:database/"
+ "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
+ "ospf:area-scope-lsa/ospf:version/ospf:ospfv2/"
+ "ospf:ospfv2/ospf:body/ospf:opaque/"
+ "ospf:ri-opaque" {
when "../../../../../../../../../../../../../../../"
+ "rt:type = 'ospf:ospfv2'" {
description
    "This augmentation is only valid for OSPFv2.";
}
description
    "Node MSD TLV is an optional TLV of OSPFv2 RI Opaque
    LSA (RFC7770) and has a type of 12.";

uses node-msd-tlv;
}

augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:database/"
+ "ospf:as-scope-lsa-type/ospf:as-scope-lsas/"
+ "ospf:as-scope-lsa/ospf:version/ospf:ospfv2/"
+ "ospf:ospfv2/ospf:body/ospf:opaque/"
+ "ospf:ri-opaque" {
when "../../../../../../../../../../../../../../../"
+ "rt:type = 'ospf:ospfv2'" {
description
    "This augmentation is only valid for OSPFv2.";
}
description
    "Node MSD TLV is an optional TLV of OSPFv2 RI Opaque
    LSA (RFC7770) and has a type of 12.";

uses node-msd-tlv;
}

augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:areas/"
+ "ospf:area/ospf:database/"

```



```

        + "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
        + "ospf:area-scope-lsa/ospf:version/ospf:ospfv3/"
        + "ospf:ospfv3/ospf:body/ospf:router-information" {
when "../.../.../.../.../.../.../.../.../..."
        + "rt:type = 'ospf:ospfv3'" {
        description
            "This augmentation is only valid for OSPFv3.";
        }
        description
            "Node MSD TLV is an optional TLV of OSPFv3 RI Opaque
            LSA (RFC7770) and has a type of 12.";

        uses node-msd-tlv;
    }

augment "/rt:routing/"
    + "rt:control-plane-protocols/rt:control-plane-protocol/"
    + "ospf:ospf/ospf:database/"
    + "ospf:as-scope-lsa-type/ospf:as-scope-lsas/"
    + "ospf:as-scope-lsa/ospf:version/ospf:ospfv3/"
    + "ospf:ospfv3/ospf:body/ospf:router-information" {
when "../.../.../.../.../.../.../.../..."
        + "rt:type = 'ospf:ospfv3'" {
        description
            "This augmentation is only valid for OSPFv3.";
        }
        description
            "Node MSD TLV is an optional TLV of OSPFv3 RI Opaque
            LSA (RFC7770) and has a type of 12.";

        uses node-msd-tlv;
    }

/* link MSD sub-tlv */
augment "/rt:routing/"
    + "rt:control-plane-protocols/rt:control-plane-protocol/"
    + "ospf:ospf/ospf:areas/ospf:area/"
    + "ospf:interfaces/ospf:interface/ospf:database/"
    + "ospf:link-scope-lsa-type/ospf:link-scope-lsas/"
    + "ospf:link-scope-lsa/ospf:version/ospf:ospfv2/"
    + "ospf:ospfv2/ospf:body/ospf:opaque/"
    + "ospf:extended-link-opaque/ospf:extended-link-tlv" {
when "../.../.../.../.../.../.../.../.../.../.../..."
        + "rt:type = 'ospf:ospfv2'" {
        description
            "This augmentation is only valid for OSPFv2.";
        }
        description

```



```

    "Link MSD sub-TLV is an optional sub-TLV of OSPFv2 extended
    link TLV as defined in RFC 7684 and has a type of 6.";

    uses link-msd-sub-tlv;
}

augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:areas/"
+ "ospf:area/ospf:database/"
+ "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
+ "ospf:area-scope-lsa/ospf:version/ospf:ospfv2/"
+ "ospf:ospfv2/ospf:body/ospf:opaque/"
+ "ospf:extended-link-opaque/ospf:extended-link-tlv" {
when "../.../.../.../.../.../.../.../.../.../..."
+ "rt:type = 'ospf:ospfv2'" {
    description
        "This augmentation is only valid for OSPFv2.";
}
description
    "Link MSD sub-TLV is an optional sub-TLV of OSPFv2 extended
    link TLV as defined in RFC 7684 and has a type of 6.";

    uses link-msd-sub-tlv;
}

augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:database/"
+ "ospf:as-scope-lsa-type/ospf:as-scope-lsas/"
+ "ospf:as-scope-lsa/ospf:version/ospf:ospfv2/"
+ "ospf:ospfv2/ospf:body/ospf:opaque/"
+ "ospf:extended-link-opaque/ospf:extended-link-tlv" {
when "../.../.../.../.../.../.../.../.../.../..."
+ "rt:type = 'ospf:ospfv2'" {
    description
        "This augmentation is only valid for OSPFv2.";
}
description
    "Link MSD sub-TLV is an optional sub-TLV of OSPFv2 extended
    link TLV as defined in RFC 7684 and has a type of 6.";

    uses link-msd-sub-tlv;
}

augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:areas/ospf:area/ospf:database/"

```



```

+ "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
+ "ospf:area-scope-lsa/ospf:version/ospf:ospfv3/"
+ "ospf:ospfv3/ospf:body/ospfv3-e-lsa:e-router"
+ "/ospfv3-e-lsa:e-router-tlvs" {
when "ospf:../..../..../..../..../..../"
+ "rt:type" = 'ospf:ospfv3' " {
description
    "This augmentation is only valid for OSPFv3
    E-Router LSAs";
}
description
    "Augment OSPFv3 Area scope router-link TLV.";

uses link-msd-sub-tlv;
}
}
<CODE ENDS>

```

3. OSPF Segment Routing

This document defines a model for OSPF Segment Routing feature [RFC8665]. It is an augmentation of the OSPF base model.

The OSPF SR YANG module requires support for the base segment routing module [RFC9020], which defines the global segment routing configuration independent of any specific routing protocol configuration, and support of OSPF base model[I-D.ietf-ospf-yang] which defines basic OSPF configuration and state.

```

module: ietf-ospf-sr
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/ospf:ospf:
+--rw segment-routing
|   +--rw enabled?      boolean
|   +--rw bindings {mapping-server}?
|       +--rw advertise
|           +--rw policies*   -> /rt:routing/sr:segment-routing
|                                   /sr-mpls:sr-mpls/bindings
|                                   /mapping-server/policy/name
|       +--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/ospf:ospf:areas/ospf:area
    /ospf:interfaces/ospf:interface:

```



```

+--rw segment-routing
  +--rw adjacency-sid
    +--rw adj-sids* [value]
      |   +--rw value-type?  enumeration
      |   +--rw value        uint32
      |   +--rw protected?   boolean
      |   +--rw weight?      uint8
      +--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/ospf:ospf/ospf:areas/ospf:area
/ospf:interfaces/ospf:interface/ospf:fast-reroute:
+--rw ti-lfa {ti-lfa}?
  +--rw enable?    boolean
augment /rt:routing/rt:control-plane-protocols
/rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
/ospf:interfaces/ospf:interface/ospf:database
/ospf:link-scope-lsa-type/ospf:link-scope-lsas
/ospf:link-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
/ospf:body/ospf:opaque/ospf:extended-prefix-opaque:
+--ro extended-prefix-range-tlvs
  +--ro extended-prefix-range-tlv* []
    +--ro prefix-length?            uint8
    +--ro af?                      uint8
    +--ro range-size?              uint16
    +--ro extended-prefix-range-flags
      |   +--ro bits*    identityref
    +--ro prefix?                inet:ip-prefix
    +--ro prefix-sid-sub-tlvs
      |   +--ro prefix-sid-sub-tlv* []
      |   |   +--ro prefix-sid-flags
      |   |   |   +--ro bits*    identityref
      |   |   +--ro mt-id?        uint8
      |   |   +--ro algorithm?    uint8
      |   |   +--ro sid?          uint32
    +--ro unknown-tlvs
      +--ro unknown-tlv* []
        +--ro type?    uint16
        +--ro length?  uint16
        +--ro value?   yang:hex-string
augment /rt:routing/rt:control-plane-protocols
/rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
/ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
/ospf:area-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
/ospf:body/ospf:opaque/ospf:extended-prefix-opaque:
+--ro extended-prefix-range-tlvs
  +--ro extended-prefix-range-tlv* []

```



```

+--ro prefix-length?                uint8
+--ro af?                           uint8
+--ro range-size?                   uint16
+--ro extended-prefix-range-flags
|   +--ro bits* identityref
+--ro prefix?                       inet:ip-prefix
+--ro prefix-sid-sub-tlvs
|   +--ro prefix-sid-sub-tlv* []
|   |   +--ro prefix-sid-flags
|   |   |   +--ro bits* identityref
|   |   +--ro mt-id?            uint8
|   |   +--ro algorithm?       uint8
|   |   +--ro sid?             uint32
+--ro unknown-tlvs
|   +--ro unknown-tlv* []
|   |   +--ro type?            uint16
|   |   +--ro length?         uint16
|   |   +--ro value?          yang:hex-string
augment /rt:routing/rt:control-plane-protocols
/rt:control-plane-protocol/ospf:ospf/ospf:database
/ospf:as-scope-lsa-type/ospf:as-scope-lsas
/ospf:as-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
/ospf:body/ospf:opaque/ospf:extended-prefix-opaque:
+--ro extended-prefix-range-tlvs
|   +--ro extended-prefix-range-tlv* []
|   |   +--ro prefix-length?    uint8
|   |   +--ro af?              uint8
|   |   +--ro range-size?      uint16
|   |   +--ro extended-prefix-range-flags
|   |   |   +--ro bits* identityref
|   |   +--ro prefix?          inet:ip-prefix
|   |   +--ro prefix-sid-sub-tlvs
|   |   |   +--ro prefix-sid-sub-tlv* []
|   |   |   |   +--ro prefix-sid-flags
|   |   |   |   |   +--ro bits* identityref
|   |   |   |   +--ro mt-id?    uint8
|   |   |   |   +--ro algorithm? uint8
|   |   |   |   +--ro sid?      uint32
|   |   +--ro unknown-tlvs
|   |   |   +--ro unknown-tlv* []
|   |   |   |   +--ro type?    uint16
|   |   |   |   +--ro length?  uint16
|   |   |   |   +--ro value?   yang:hex-string
augment /rt:routing/rt:control-plane-protocols
/rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
/ospf:interfaces/ospf:interface/ospf:database
/ospf:link-scope-lsa-type/ospf:link-scope-lsas
/ospf:link-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2

```



```

        /ospf:body/ospf:opaque/ospf:extended-prefix-opaque
        /ospf:extended-prefix-tlv:
+---ro prefix-sid-sub-tlvs
  +---ro prefix-sid-sub-tlv* []
    +---ro prefix-sid-flags
      | +---ro bits* identityref
    +---ro mt-id? uint8
    +---ro algorithm? uint8
    +---ro sid? uint32
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
  /ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
  /ospf:area-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
  /ospf:body/ospf:opaque/ospf:extended-prefix-opaque
  /ospf:extended-prefix-tlv:
+---ro prefix-sid-sub-tlvs
  +---ro prefix-sid-sub-tlv* []
    +---ro prefix-sid-flags
      | +---ro bits* identityref
    +---ro mt-id? uint8
    +---ro algorithm? uint8
    +---ro sid? uint32
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/ospf:ospf/ospf:database
  /ospf:as-scope-lsa-type/ospf:as-scope-lsas/ospf:as-scope-lsa
  /ospf:version/ospf:ospfv2/ospf:ospfv2/ospf:body/ospf:opaque
  /ospf:extended-prefix-opaque/ospf:extended-prefix-tlv:
+---ro prefix-sid-sub-tlvs
  +---ro prefix-sid-sub-tlv* []
    +---ro prefix-sid-flags
      | +---ro bits* identityref
    +---ro mt-id? uint8
    +---ro algorithm? uint8
    +---ro sid? uint32
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
  /ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
  /ospf:area-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
  /ospf:body/ospf:opaque/ospf:extended-link-opaque
  /ospf:extended-link-tlv:
+---ro adj-sid-sub-tlvs
  | +---ro adj-sid-sub-tlv* []
  | +---ro adj-sid-flags
  | | +---ro bits* identityref
  | +---ro mt-id? uint8
  | +---ro weight? uint8
  | +---ro sid? uint32
+---ro lan-adj-sid-sub-tlvs

```



```

    +---ro lan-adj-sid-sub-tlv* []
    +---ro lan-adj-sid-flags
    |   +---ro bits*      identityref
    +---ro mt-id?          uint8
    +---ro weight?         uint8
    +---ro neighbor-router-id?  yang:dotted-quad
    +---ro sid?            uint32
augment /rt:routing/rt:control-plane-protocols
/rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
/ospf:interfaces/ospf:interface/ospf:database
/ospf:link-scope-lsa-type/ospf:link-scope-lsas
/ospf:link-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
/ospf:body/ospf:opaque/ospf:ri-opaque:
+---ro sr-algorithm-tlv
|   +---ro sr-algorithm*   uint8
+---ro sid-range-tlvs
|   +---ro sid-range-tlv* []
|   +---ro range-size?    uint24
|   +---ro sid-sub-tlv
|   |   +---ro sid?      uint32
+---ro local-block-tlvs
|   +---ro local-block-tlv* []
|   +---ro range-size?    uint24
|   +---ro sid-sub-tlv
|   |   +---ro sid?      uint32
+---ro srms-preference-tlv
    +---ro preference?    uint8
augment /rt:routing/rt:control-plane-protocols
/rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
/ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
/ospf:area-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
/ospf:body/ospf:opaque/ospf:ri-opaque:
+---ro sr-algorithm-tlv
|   +---ro sr-algorithm*   uint8
+---ro sid-range-tlvs
|   +---ro sid-range-tlv* []
|   +---ro range-size?    uint24
|   +---ro sid-sub-tlv
|   |   +---ro sid?      uint32
+---ro local-block-tlvs
|   +---ro local-block-tlv* []
|   +---ro range-size?    uint24
|   +---ro sid-sub-tlv
|   |   +---ro sid?      uint32
+---ro srms-preference-tlv
    +---ro preference?    uint8
augment /rt:routing/rt:control-plane-protocols
/rt:control-plane-protocol/ospf:ospf/ospf:database

```



```

        /ospf:as-scope-lsa-type/ospf:as-scope-lsas/ospf:as-scope-lsa
        /ospf:version/ospf:ospfv2/ospf:ospfv2/ospf:body/ospf:opaque
        /ospf:ri-opaque:
+--ro sr-algorithm-tlv
|   +--ro sr-algorithm*      uint8
+--ro sid-range-tlvs
|   +--ro sid-range-tlv* []
|       +--ro range-size?    uint24
|       +--ro sid-sub-tlv
|           +--ro sid?      uint32
+--ro local-block-tlvs
|   +--ro local-block-tlv* []
|       +--ro range-size?    uint24
|       +--ro sid-sub-tlv
|           +--ro sid?      uint32
+--ro srms-preference-tlv
    +--ro preference?      uint8

```

3.1. OSPF Segment Routing YANG Module

```

<CODE BEGINS> file "ietf-ospf-sr@2022-01-02.yang"
module ietf-ospf-sr {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-ospf-sr";

  prefix ospf-sr;

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

  import ietf-segment-routing-common {
    prefix "sr-cmn";
    reference "RFC 9020 - YANG Data Model for Segment
              Routing";
  }

```



```
}
import ietf-segment-routing-mpls {
  prefix "sr-mpls";
  reference "RFC 9020 - YANG Data Model for Segment
    Routing";
}
import ietf-ospf {
  prefix "ospf";
}

organization
  "IETF LSR - Link State Routing Working Group";

contact
  "WG Web:    <http://tools.ietf.org/wg/lsr/>
  WG List:    <mailto:lsr@ietf.org>

  Editor:     Derek Yeung
               <mailto:derek@arccus.com>
  Author:     Derek Yeung
               <mailto:derek@arccus.com>
  Author:     Yingzhen Qu
               <mailto:yingzhen.qu@futurewei.com>
  Author:     Acee Lindem
               <mailto:acee@cisco.com>
  Author:     Jeffrey Zhang
               <mailto:zzhang@juniper.net>
  Author:     Ing-Wher Chen
               <mailto:ingwherchen@mitre.org>
  Author:     Greg Hankins
               <mailto:greg.hankins@alcatel-lucent.com>";

description
  "This YANG module defines the generic configuration
  and operational state for OSPF Segment Routing, which is
  common across all of the vendor implementations. It is
  intended that the module will be extended by vendors to
  define vendor-specific OSPF Segment Routing configuration
  and operational parameters and policies.

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

  Copyright (c) 2022 IETF Trust and the persons identified as
  authors of the code. All rights reserved.

  Redistribution and use in source and binary forms, with or
  without modification, is permitted pursuant to, and subject to
```


the license terms contained in, the Revised BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC 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 2022-01-02 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: A YANG Data Model for OSPF Segment Routing.";
}
```

```
feature ti-lfa {
  description
    "Topology-Independent Loop-Free Alternate (TI-LFA)
    computation using segment routing.";
}
```

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

```
identity np-bit {
  base prefix-sid-bit;
  description
    "No-PHP flag.";
}
```

```
identity m-bit {
  base prefix-sid-bit;
  description
```



```
    "Mapping server flag.";
  }

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

  identity v-bit {
    base prefix-sid-bit;
    description
      "Value/Index flag.";
  }

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

  identity extended-prefix-range-bit {
    description
      "Base identity for extended prefix range TLV bits.";
  }

  identity ia-bit {
    base extended-prefix-range-bit;
    description
      "Inter-Area flag. If set, advertisement is of inter-area type.";
  }

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

  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/Global flag.";
}

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

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

typedef uint24 {
    type uint32 {
        range "0 .. 16777215";
    }
    description
        "24-bit unsigned integer.";
}

/* 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 prefix-sid-sub-tlvs {
    description "Prefix Segment ID (SID) sub-TLVs.";
    container prefix-sid-sub-tlvs {
        description "Prefix SID sub-TLV.";
        list prefix-sid-sub-tlv {
            description "Prefix SID sub-TLV.";
        }
    }
}
```



```
    container prefix-sid-flags {
      leaf-list bits {
        type identityref {
          base prefix-sid-bit;
        }
        description
          "Prefix SID Sub-TLV flag bits list.";
      }
      description "Segment Identifier (SID) Flags.";
    }
    leaf mt-id {
      type uint8;
      description "Multi-topology ID.";
    }
    leaf algorithm {
      type uint8;
      description
        "The algorithm associated with the prefix-SID.";
    }
    leaf sid {
      type uint32;
      description "An index or label.";
    }
  }
}

grouping extended-prefix-range-tlvs {
  description "Extended prefix range TLV grouping.";

  container extended-prefix-range-tlvs {
    description "The list of range of prefixes.";
    list extended-prefix-range-tlv {
      description "The range of prefixes.";
      leaf prefix-length {
        type uint8;
        description "Length of prefix in bits.";
      }
      leaf af {
        type uint8;
        description "Address family for the prefix.";
      }
      leaf range-size {
        type uint16;
        description "The number of prefixes covered by the
          advertisement.";
      }
      container extended-prefix-range-flags {
```



```

        leaf-list bits {
            type identityref {
                base extended-prefix-range-bit;
            }
            description "Extended prefix range TLV flags list.";
        }
        description "Extended Prefix Range TLV flags.";
    }
    leaf prefix {
        type inet:ip-prefix;
        description "Address prefix.";
    }
    uses prefix-sid-sub-tlvs;
    uses ospf:unknown-tlvs;
}
}

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

grouping sid-range-tlvs {
    description "SID Range TLV grouping.";
    container sid-range-tlvs {
        description "List of SID range TLVs.";
        list sid-range-tlv {
            description "SID range TLV.";
            leaf range-size {
                type uint24;
                description "The SID range.";
            }
            uses sid-sub-tlv;
        }
    }
}

grouping local-block-tlvs {
    description "The SR local block TLV contains the

```



```

        range of labels reserved for local SIDs.";
    container local-block-tlvs {
        description "List of SRLB TLVs.";
        list local-block-tlv {
            description "SRLB TLV.";
            leaf range-size {
                type uint24;
                description "The SID range.";
            }
            uses sid-sub-tlv;
        }
    }
}

grouping srms-preference-tlv {
    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-tlv {
        description "SRMS Preference TLV.";
        leaf preference {
            type uint8 {
                range "0 .. 255";
            }
            description "SRMS preference TLV, value from 0 to 255.";
        }
    }
}

/* Configuration */
augment "/rt:routing/rt:control-plane-protocols"
    + "/rt:control-plane-protocol/ospf:ospf" {
    when "../rt:type = 'ospf:ospfv2' or "
    + "../rt:type = 'ospf:ospfv3'" {
        description
            "This augments the OSPF routing protocol when used.";
    }
    description
        "This augments the OSPF protocol configuration
        with segment routing.";
    uses sr-mpls:sr-control-plane;
    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/ospf:ospf/"
  + "ospf:areas/ospf:area/ospf:interfaces/ospf:interface" {
when "../.../.../rt:type = 'ospf:ospfv2' or "
  + "../.../.../rt:type = 'ospf:ospfv3'" {
  description
    "This augments the OSPF interface configuration
    when used.";
}
description
  "This augments the OSPF protocol interface
  configuration with segment routing.";

  uses sr-mpls:igp-interface;
}

augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/ospf:ospf/"
  + "ospf:areas/ospf:area/ospf:interfaces/ospf:interface/"
  + "ospf:fast-reroute" {
when "../.../.../rt:type = 'ospf:ospfv2' or "
  + "../.../.../rt:type = 'ospf:ospfv3'" {
  description
    "This augments the OSPF routing protocol when used.";
}
description
  "This augments the OSPF protocol IP-FRR with TI-LFA.";

  container ti-lfa {
    if-feature ti-lfa;
    leaf enable {
      type boolean;
      description
        "Enables TI-LFA computation.";
    }
    description
      "Topology Independent Loop Free Alternate
      (TI-LFA) support.";
  }
}

/* Database */
augment "/rt:routing/"
  + "rt:control-plane-protocols/rt:control-plane-protocol/"
  + "ospf:ospf/ospf:areas/ospf:area/"
  + "ospf:interfaces/ospf:interface/ospf:database/"
  + "ospf:link-scope-lsa-type/ospf:link-scope-lsas/"
  + "ospf:link-scope-lsa/ospf:version/ospf:ospfv2/"

```



```

        + "ospf:ospfv2/ospf:body/ospf:opaque/"
        + "ospf:extended-prefix-opaque" {
when "../.../.../.../.../.../.../.../.../.../.../..."
    + "rt:type = 'ospf:ospfv2'" {
        description
            "This augmentation is only valid for OSPFv2.";
    }
    description
        "SR specific TLVs for OSPFv2 extended prefix LSA
        in type 9 opaque LSA.";

    uses extended-prefix-range-tlvs;
}

augment "/rt:routing/"
    + "rt:control-plane-protocols/rt:control-plane-protocol/"
    + "ospf:ospf/ospf:areas/"
    + "ospf:area/ospf:database/"
    + "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
    + "ospf:area-scope-lsa/ospf:version/ospf:ospfv2/"
    + "ospf:ospfv2/ospf:body/ospf:opaque/"
    + "ospf:extended-prefix-opaque" {
when "../.../.../.../.../.../.../.../.../.../..."
    + "rt:type = 'ospf:ospfv2'" {
        description
            "This augmentation is only valid for OSPFv2.";
    }
    description
        "SR specific TLVs for OSPFv2 extended prefix LSA
        in type 10 opaque LSA.";

    uses extended-prefix-range-tlvs;
}

augment "/rt:routing/"
    + "rt:control-plane-protocols/rt:control-plane-protocol/"
    + "ospf:ospf/ospf:database/"
    + "ospf:as-scope-lsa-type/ospf:as-scope-lsas/"
    + "ospf:as-scope-lsa/ospf:version/ospf:ospfv2/"
    + "ospf:ospfv2/ospf:body/ospf:opaque/"
    + "ospf:extended-prefix-opaque" {
when "../.../.../.../.../.../.../.../.../.../..."
    + "rt:type = 'ospf:ospfv2'" {
        description
            "This augmentation is only valid for OSPFv2.";
    }
    description
        "SR specific TLVs for OSPFv2 extended prefix LSA

```



```

        in type 11 opaque LSA.";

    uses extended-prefix-range-tlvs;
}

augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:areas/ospf:area/"
+ "ospf:interfaces/ospf:interface/ospf:database/"
+ "ospf:link-scope-lsa-type/ospf:link-scope-lsas/"
+ "ospf:link-scope-lsa/ospf:version/ospf:ospfv2/"
+ "ospf:ospfv2/ospf:body/ospf:opaque/"
+ "ospf:extended-prefix-opaque/ospf:extended-prefix-tlv" {
when "../.../.../.../.../.../.../.../.../.../..."
+ "rt:type = 'ospf:ospfv2'" {
    description
        "This augmentation is only valid for OSPFv2.";
}
description
    "SR specific TLVs for OSPFv2 extended prefix TLV
    in type 9 opaque LSA.";
uses prefix-sid-sub-tlvs;
}

augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:areas/"
+ "ospf:area/ospf:database/"
+ "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
+ "ospf:area-scope-lsa/ospf:version/ospf:ospfv2/"
+ "ospf:ospfv2/ospf:body/ospf:opaque/"
+ "ospf:extended-prefix-opaque/ospf:extended-prefix-tlv" {
when "../.../.../.../.../.../.../.../.../.../..."
+ "rt:type = 'ospf:ospfv2'" {
    description
        "This augmentation is only valid for OSPFv2.";
}
description
    "SR specific TLVs for OSPFv2 extended prefix TLV
    in type 10 opaque LSA.";
uses prefix-sid-sub-tlvs;
}

augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:database/"
+ "ospf:as-scope-lsa-type/ospf:as-scope-lsas/"
+ "ospf:as-scope-lsa/ospf:version/ospf:ospfv2/"

```



```

    + "ospf:ospfv2/ospf:body/ospf:opaque/"
    + "ospf:extended-prefix-opaque/ospf:extended-prefix-tlv" {
when "../.../.../.../.../.../.../.../.../.../..."
    + "rt:type = 'ospf:ospfv2'" {
    description
        "This augmentation is only valid for OSPFv2.";
    }
    description
        "SR specific TLVs for OSPFv2 extended prefix TLV
        in type 11 opaque LSA.";
    uses prefix-sid-sub-tlvs;
}

augment "/rt:routing/"
    + "rt:control-plane-protocols/rt:control-plane-protocol/"
    + "ospf:ospf/ospf:areas/"
    + "ospf:area/ospf:database/"
    + "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
    + "ospf:area-scope-lsa/ospf:version/ospf:ospfv2/"
    + "ospf:ospfv2/ospf:body/ospf:opaque/"
    + "ospf:extended-link-opaque/ospf:extended-link-tlv" {
when "../.../.../.../.../.../.../.../.../.../..."
    + "rt:type = 'ospf:ospfv2'" {
    description
        "This augmentation is only valid for OSPFv2.";
    }
    description
        "SR specific TLVs for OSPFv2 extended link TLV
        in type 10 opaque LSA.";

    container adj-sid-sub-tlvs {
        description "Adjacency SID optional sub-TLVs.";
        list adj-sid-sub-tlv {
            description "List of Adjacency SID sub-TLVs.";
            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 mt-id {
                type uint8;
                description "Multi-topology ID.";
            }
            leaf weight {

```



```

        type uint8;
        description "Weight used for load-balancing.";
    }
    leaf sid {
        type uint32;
        description "Segment Identifier (SID) index/label.";
    }
}

container lan-adj-sid-sub-tlvs {
    description "LAN Adjacency SID optional sub-TLVs.";
    list lan-adj-sid-sub-tlv {
        description "List of LAN adjacency SID sub-TLVs.";
        container lan-adj-sid-flags {
            leaf-list bits {
                type identityref {
                    base adj-sid-bit;
                }
                description "LAN adj sid sub-tlv flags list.";
            }
            description "LAN adj-sid sub-tlv flags.";
        }
        leaf mt-id {
            type uint8;
            description "Multi-topology ID.";
        }
        leaf weight {
            type uint8;
            description "Weight used for load-balancing.";
        }
        leaf neighbor-router-id {
            type yang:dotted-quad;
            description "Neighbor router ID.";
        }
        leaf sid {
            type uint32;
            description "Segment Identifier (SID) index/label.";
        }
    }
}

augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:areas/ospf:area/"
+ "ospf:interfaces/ospf:interface/ospf:database/"
+ "ospf:link-scope-lsa-type/ospf:link-scope-lsas/"

```



```

        + "ospf:link-scope-lsa/ospf:version/ospf:ospfv2/"
        + "ospf:ospfv2/ospf:body/ospf:opaque/ospf:ri-opaque" {
when "../.../.../.../.../.../.../.../.../.../.../..."
    + "rt:type = 'ospf:ospfv2'" {
        description
            "This augmentation is only valid for OSPFv2.";
    }

description
    "SR specific TLVs for OSPFv2 type 9 opaque LSA.";

uses sr-algorithm-tlv;
uses sid-range-tlvs;
uses local-block-tlvs;
uses srms-preference-tlv;
}

augment "/rt:routing/"
    + "rt:control-plane-protocols/rt:control-plane-protocol/"
    + "ospf:ospf/ospf:areas/"
    + "ospf:area/ospf:database/"
    + "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
    + "ospf:area-scope-lsa/ospf:version/ospf:ospfv2/"
    + "ospf:ospfv2/ospf:body/ospf:opaque/ospf:ri-opaque" {
when "../.../.../.../.../.../.../.../.../.../..."
    + "rt:type = 'ospf:ospfv2'" {
        description
            "This augmentation is only valid for OSPFv2.";
    }

description
    "SR specific TLVs for OSPFv2 type 10 opaque LSA.";

uses sr-algorithm-tlv;
uses sid-range-tlvs;
uses local-block-tlvs;
uses srms-preference-tlv;
}

augment "/rt:routing/"
    + "rt:control-plane-protocols/rt:control-plane-protocol/"
    + "ospf:ospf/ospf:database/"
    + "ospf:as-scope-lsa-type/ospf:as-scope-lsas/"
    + "ospf:as-scope-lsa/ospf:version/ospf:ospfv2/"
    + "ospf:ospfv2/ospf:body/ospf:opaque/ospf:ri-opaque" {
when "../.../.../.../.../.../.../.../.../..."
    + "rt:type = 'ospf:ospfv2'" {
        description

```



```

        "This augmentation is only valid for OSPFv2.";
    }
    description
        "SR specific TLVs for OSPFv2 type 11 opaque LSA.";

    uses sr-algorithm-tlv;
    uses sid-range-tlvs;
    uses local-block-tlvs;
    uses srms-preference-tlv;
}
}
<CODE ENDS>

```

4. 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 Configuration 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 the modules that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

/ospf:ospf/segment-routing/enabled - Modification to the enablement for SR could result in a Denial-of-Service (Dos) attack. If an attacker disables SR, it will cause traffic disruption.

/ospf:ospf/segment-routing/bindings - Modification to the local bindings could result in a Denial-of-Service (Dos) attack.

/ospf:ospf/protocol-srgb - Modification of the protocol SRGB could be used to mount a DoS attack. For example, if the protocol SRBG size is reduced to a very small value, a lot of existing segments could no longer be installed leading to a traffic disruption.

/ospf:interfaces/ospf:interface/segment-routing - Modification of the Adjacency Segment Identifier (Adj-SID) could be used to mount a DoS attack. Change of an Adj-SID could be used to redirect traffic.

/ospf:interfaces/ospf:interface/ospf:fast-reroute/ti-lfa - Modification of the TI-LFA enablement could lead to traffic disruption.

Some of the readable data nodes in the modules 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.

Both module ietf-ospf-sr and ietf-ospf-msd augment base OSPF module data base with various TLVs. Knowledge of these data nodes can be used to attack other routers in the OSPF domain.

5. Acknowledgements

The authors wish to thank Yi Yang, Alexander Clemm, Gaurav Gupta, Ladislav Lhotka, Stephane Litkowski, Greg Hankins, Manish Gupta and Alan Davey for their thorough reviews and helpful comments.

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

Author affiliation with The MITRE Corporation is provided for identification purposes only, and is not intended to convey or imply MITRE's concurrence with, or support for, the positions, opinions or viewpoints expressed. MITRE has approved this document for Public Release, Distribution Unlimited, with Public Release Case Number 18-3281.

6. IANA Considerations

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

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

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

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

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

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

7. References

7.1. Normative References

- [I-D.ietf-ospf-yang]
Yeung, D., Qu, Y., Zhang, J., Chen, I., and A. Lindem,
"YANG Data Model for OSPF Protocol", Work in Progress,
Internet-Draft, draft-ietf-ospf-yang-29, 17 October 2019,
<<https://www.ietf.org/archive/id/draft-ietf-ospf-yang-29.txt>>.
- [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>>.
- [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328,
DOI 10.17487/RFC2328, April 1998,
<<https://www.rfc-editor.org/info/rfc2328>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688,
DOI 10.17487/RFC3688, January 2004,
<<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC4750] Joyal, D., Ed., Galecki, P., Ed., Giacalone, S., Ed.,
Coltun, R., and F. Baker, "OSPF Version 2 Management
Information Base", RFC 4750, DOI 10.17487/RFC4750,
December 2006, <<https://www.rfc-editor.org/info/rfc4750>>.
- [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF
for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008,
<<https://www.rfc-editor.org/info/rfc5340>>.

- [RFC5643] Joyal, D., Ed. and V. Manral, Ed., "Management Information Base for OSPFv3", RFC 5643, DOI 10.17487/RFC5643, August 2009, <<https://www.rfc-editor.org/info/rfc5643>>.
- [RFC5838] Lindem, A., Ed., Mirtorabi, S., Roy, A., Barnes, M., and R. Aggarwal, "Support of Address Families in OSPFv3", RFC 5838, DOI 10.17487/RFC5838, April 2010, <<https://www.rfc-editor.org/info/rfc5838>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 7223, DOI 10.17487/RFC7223, May 2014, <<https://www.rfc-editor.org/info/rfc7223>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [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>>.
- [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>>.
- [RFC8476] Tantsura, J., Chunduri, U., Aldrin, S., and P. Psenak, "Signaling Maximum SID Depth (MSD) Using OSPF", RFC 8476, DOI 10.17487/RFC8476, December 2018, <<https://www.rfc-editor.org/info/rfc8476>>.

- [RFC8665] Psenak, P., Ed., Previdi, S., Ed., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", RFC 8665, DOI 10.17487/RFC8665, December 2019, <<https://www.rfc-editor.org/info/rfc8665>>.
- [RFC9020] Litkowski, S., Qu, Y., Lindem, A., Sarkar, P., and J. Tantsura, "YANG Data Model for Segment Routing", RFC 9020, DOI 10.17487/RFC9020, May 2021, <<https://www.rfc-editor.org/info/rfc9020>>.

7.2. Informative References

- [RFC8022] Lhotka, L. and A. Lindem, "A YANG Data Model for Routing Management", RFC 8022, DOI 10.17487/RFC8022, November 2016, <<https://www.rfc-editor.org/info/rfc8022>>.
- [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>>.

Appendix A. Contributors' Addreses

Dean Bogdanovic
Volta Networks, Inc.

EMail: dean@voltanet.io

Kiran Koushik Agrahara Sreenivasa
Cisco Systems
12515 Research Blvd, Bldg 4
Austin, TX 78681
USA

EMail: kkoushik@cisco.com

Authors' Addresses

Derek Yeung
Arrcus

Email: derek@arrcus.com

Yingzhen Qu
Futurewei
2330 Central Expressway
Santa Clara, CA 95050
United States of America

Email: yingzhen.qu@futurewei.com

Jeffrey Zhang
Juniper Networks
10 Technology Park Drive
Westford, MA 01886
United States of America

Email: zzhang@juniper.net

Ing-Wher Chen
The MITRE Corporation

Email: ingwherchen@mitre.org

Acee Lindem
Cisco Systems
301 Midenhall Way
Cary, NC 27513

Email: acee@cisco.com

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: April 5, 2020

T. Li
S. Chen
Arista Networks
October 3, 2019

Area Proxy for IS-IS
draft-li-lsr-isis-area-proxy-00

Abstract

Link state routing protocols have hierarchical abstraction already built into them. However, when lower levels are used for transit, they must expose their internal topologies to each other, leading to scale issues.

To avoid this, this document discusses extensions to the IS-IS routing protocol that would allow level 1 areas to provide transit, yet only inject an abstraction of the level 1 topology into level 2. Each level 1 area is represented as a single level 2 node, thereby enabling greater scale.

Status of This Memo

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

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

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

This Internet-Draft will expire on April 5, 2020.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Requirements Language	3
2. Area Proxy	4
3. Inside Router Functions	5
3.1. The Area Proxy Router Capability	5
3.2. Level 2 SPF Computation	5
4. Area Leader Functions	6
4.1. Area Leader Election	6
4.2. Redundancy	6
4.3. Area Proxy System Identifier TLV	6
4.4. Area Proxy LSP Generation	7
5. Inside Edge Router Functions	8
5.1. Generating L2 IIHs to Outside Routers	8
5.2. Filtering LSP information	8
6. Acknowledgments	9
7. IANA Considerations	9
8. Security Considerations	9
9. References	9
9.1. Normative References	9
9.2. Informative References	10
9.3. URIs	10
Authors' Addresses	10

1. Introduction

The IS-IS routing protocol IS-IS [ISO10589] currently supports a two-level hierarchy of abstraction. The fundamental unit of abstraction is the 'area', which is a (hopefully) connected set of systems running IS-IS at the same level. Level 1, the lowest level, is abstracted by routers that participate in both Level 1 and Level 2, and they inject area information into Level 2. Level 2 systems seeking to access Level 1, use this abstraction to compute the shortest path to the Level 1 area. The full topology database of Level 1 is not injected into Level 2, only a summary of the address space contained within the area, so the scalability of the Level 2 Link State Database (LSDB) is protected.

This works well if the Level 1 area is tangential to the Level 2 area. This also works well if there are several routers in both Level 1 and Level 2 and they are adjacent, so Level 2 traffic will

never need to transit Level 1 only routers. Level 1 will not contain any Level 2 topology, and Level 2 will only contain area abstractions for Level 1.

Unfortunately, this scheme does not work so well if the Level 1 only area needs to provide transit for Level 2 traffic. For Level 2 shortest path first (SPF) computations to work correctly, the transit topology must also appear in the Level 2 LSDB. This implies that all routers that could provide transit, plus any links that might also provide Level 2 transit must also become part of the Level 2 topology. If this is a relatively tiny portion of the Level 1 area, this is not overly painful.

However, with today's data center topologies, this is problematic. A common application is to use a Layer 3 Leaf-Spine (L3LS) topology, which is a folded 3-stage Clos [Clos] fabric. It can also be thought of as a complete bipartite graph. In such a topology, the desire is to use Level 1 to contain the routing dynamics of the entire L3LS topology and then to use Level 2 for the remainder of the network. Leaves in the L3LS topology are appropriate for connection outside of the data center itself, so they would provide connectivity for Level 2. If there are multiple connections to Level 2 for redundancy, or other areas, these too would also be made to the leaves in the topology. This creates a difficulty because there are now multiple Level 2 leaves in the topology, with connectivity between the leaves provided by the spines.

Following the current rules of IS-IS, all spine routers would necessarily be part of the Level 2 topology, plus all links between a Level 2 leaf and the spines. In the limit, where all leaves need to support Level 2, it implies that the entire L3LS topology becomes part of Level 2. This is seriously problematic as it more than doubles the LSDB held in the L3LS topology and eliminates any benefits of the hierarchy.

This document discusses the handling of IP traffic. Supporting MPLS based traffic is a subject for future work.

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 BCP 14 [1] [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Area Proxy

To address this, we propose to completely abstract away the details of the Level 1 area topology within Level 2, making the entire area look like a single proxy system directly connected to all of the area's Level 2 neighbors. By only providing an abstraction of the topology, Level 2's requirement for connectivity can be satisfied without the full overhead of the area's internal topology. It then becomes the responsibility of the Level 1 area to ensure the forwarding connectivity that's advertised.

For this discussion, we'll consider a single Level 1 IS-IS area to be the Inside Area, and the remainder of the Level 2 area is the Outside Area. All routers within the Inside Area speak Level 1 and Level 2 IS-IS on all of the links within the topology. We propose to implement Area Proxy by having a Level 2 Proxy Link State Protocol Data Unit (PDU, LSP) that represents the entire Inside Area. This is the only LSP from the area that will be flooded into the overall Level 2 LSDB.

There are four classes of routers that we need to be concerned with in this discussion:

Inside Router A router within the Inside Area that runs Level 1 and Level 2 IS-IS.

Area Leader The Area Leader is an Inside Router that is elected to represent the Level 1 area by injecting the Proxy LSP into the Level 2 LSDB. There may be multiple candidates for Area Leader, but only one is elected at a given time.

Inside Edge Router An Inside Edge Router is an Inside Area Router that has at least one Level 2 interface outside of the Inside Area.

Outside Edge Router An Outside Edge Router is a Level 2 router that is outside of the Inside Area that has an adjacency with an Inside Edge Router.

All Inside Edge Routers learn the Area Proxy System Identifier from the Level 1 LSDB and use that as the system identifier in their Level 2 IS-IS Hello PDUs (IIHs) on all Outside interfaces. Outside Edge Routers should then advertise an adjacency to the Area Proxy System Identifier. This allows all Outside Routers to use the Proxy LSP in their SPF computations without seeing the full topology of the Inside Area.

Area Proxy functionality assumes that all circuits are either Level 1-2 circuits within the Inside Area, or Level 2 circuits between Outside Routers and a single Inside Edge Router. Multi-access circuits (i.e. Ethernet in LAN mode) with multiple Inside Edge Routers and an Outside Router are not supported.

3. Inside Router Functions

All Inside Routers run Level 1-2 IS-IS and must be explicitly instructed to enable the Area Proxy functionality. To signal their readiness to participate in Area Proxy functionality, they will advertise the Area Proxy Router Capability as part of its Level 1 Router Capability TLV.

3.1. The Area Proxy Router Capability

The Area Proxy Router Capability is a sub-TLV of the Router Capability TLV [RFC7981] and has the following format:

```

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

```

TLV Type: YYY

TLV Length: 0

A router advertising this TLV indicates that it is running Level 1-2 and is prepared to perform Area Proxy functions.

3.2. Level 2 SPF Computation

When Outside Routers perform a Level 2 SPF computation, they will use the Area Proxy LSP for computing a path transiting the Inside Area. Because the topology has been abstracted away, the cost for transiting the Inside Area will be zero.

When Inside Routers perform a Level 2 SPF computation, they must ignore the Area Proxy LSP. Further, because these systems do see the Inside Area topology, the link metrics internal to the area are visible. This could lead to different and possibly inconsistent SPF results, potentially leading to forwarding loops.

To prevent this, the Inside Routers must consider the metrics of links outside of the Inside Area (inter-area metrics) separately from the metrics of the Inside Area links (intra-area metrics). Intra-

area metrics are always less than any inter-area metric. Thus, if two paths have different total inter-area metrics, the path with the lower inter-area metric would be preferred, regardless of any intra-area metrics involved. However, if two paths have equal inter-area metrics, then the intra-area metrics would be used to compare the paths.

4. Area Leader Functions

The Area Leader has several responsibilities. First, it must inject the Area Proxy System Identifier into the Level 1 LSDB. Second, the Area Leader must generate the Proxy LSP for the Inside Area.

4.1. Area Leader Election

The Area Leader is selected using the election mechanisms and TLVs described in Dynamic Flooding for IS-IS [I-D.ietf-lsr-dynamic-flooding].

4.2. Redundancy

If the Area Leader fails, another candidate may become Area Leader and MUST regenerate the Area Proxy LSP. The failure of the Area Leader is not visible outside of the area and appears to simply be an update of the Area Proxy LSP.

4.3. Area Proxy System Identifier TLV

The Area Proxy System Identifier TLV allows the Area Leader to advertise the existence of an Area Proxy System Identifier. This TLV is injected into the Area Leader's Level 1 LSP.

The format of the Area Proxy System Identifier TLV is:

0										1										2									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3						
TLV Type										TLV Length										Proxy SysID									
Proxy System Identifier continued ...																													

TLV Type: XXX

TLV Length: length of a system ID (6)

Proxy System Identifier: the Area Proxy System Identifier.

The Area Leader MAY advertise the Area Proxy System Identifier TLV when it observes that all Inside Routers are advertising the Area Proxy Router Capability. Their advertisements indicate that they are individually ready to perform Area Proxy functionality. The Area Leader then advertises the Area Proxy System Identifier TLV to indicate that the Inside Area should enable Area Proxy functionality.

Other candidates for Area Leader MAY also advertise the Area Proxy System Identifier when they observe that all Inside Routers are advertising the Area Proxy Router Capability. All candidates advertising the Area Proxy System Identifier TLV MUST be advertising the same system identifier. Multiple proxy system identifiers in a single area is a misconfiguration.

The Area Leader and other candidates for Area Leader MAY withdraw the Area Proxy System Identifier when one or more Inside Routers are not advertising the Area Proxy Router Capability. This will disable Area Proxy functionality. However, before withdrawing the Area Proxy System Identifier, an implementation should protect against unnecessary churn from transients by delaying the withdrawal. The amount of delay is implementation-dependent.

4.4. Area Proxy LSP Generation

Each Inside Router generates a Level 2 LSP, and the Level 2 LSPs for the Inside Edge Routers will include adjacencies to Outside Edge Routers. Unlike normal Level 2 operations, these LSPs are not advertised outside of the Inside Area and must be filtered by all Inside Edge Routers to not be flooded to Outside Routers.

The Area Leader uses the Level 2 LSPs generated by the Inside Edge Routers to generate the Area Proxy LSP. This LSP is originated using the Area Proxy System Identifier and includes adjacencies for all of the Outside Edge Routers that have been advertised by the Inside Edge Routers. Since the Outside Edge Routers also advertise an adjacency to the proxy system identifier, this will result in a bi-directional adjacency. The Area Proxy LSP is the only LSP that is injected into the overall Level 2 LSDB, with all other Level 2 LSPs from the Inside Area being filtered out at the Inside Area boundary.

The Area Leader may also insert additional TLVs into the Area Proxy LSP for additional information for the Outside Area. It is RECOMMENDED that the Area Leader insert the Dynamic Hostname TLV [RFC5301] into the Area Proxy LSP. The Area Leader SHOULD insert additional TLVs describing any routing prefixes that should be advertised on behalf of the area. These prefixes may be learned from the Level 1 LSDB, statically configured, or redistributed from

another routing protocol, using the usual TLVs for prefix advertisement. [RFC5305] [RFC5308] [RFC5120]

5. Inside Edge Router Functions

The Inside Edge Router has two additional and important functions. First, it must generate IIHs that appear to have come from the Area Proxy System Identifier. Second, it must filter the L2 LSPs, Partial Sequence Number PDUs (PSNPs), and Complete Sequence Number PDUs (CSNPs) that are being advertised to Outside Routers.

5.1. Generating L2 IIHs to Outside Routers

The Inside Edge Router has one or more Level 2 interfaces to Outside Routers. These may be identified by explicit configuration or by the fact that they are not also Level 1 circuits. On these Level 2 interfaces, the Inside Edge Router MUST NOT send an IIH until it has learned the Area Proxy System Id from the Area Leader. Then, once it has learned the Area Proxy System Id, it should generate its IIHs on the circuit using the Proxy System Id as the source of the IIH.

Using the Proxy System Id causes the Outside Router to advertise an adjacency to the Proxy System Id, not to the Inside Edge Router, which supports the proxy function. The normal system id of the Inside Edge Router MUST NOT be used as it will cause unnecessary adjacencies to form and subsequently flap.

5.2. Filtering LSP information

For the proxy abstraction to be effective the L2 LSPs generated by the Inside Routers MUST be restricted to the Inside Area. The Inside Routers know which system ids are members of the Inside Area based on the Level 1 LSDB. To prevent unwanted LSP information from escaping the Inside Area, the Inside Edge Router MUST perform filtering of LSP flooding, CSNPs, and PSNPs. Specifically:

A Level 2 LSP with a source system identifier that is found in the Level 1 LSDB should never be flooded to an Outside Router.

A Level 2 CSNP sent to an Outside Router MUST NOT contain any information about an LSP with a system identifier found in the Level 1 LSDB. If an Inside Edge Router filters a CSNP and there is no remaining content, then the CSNP MUST NOT be sent. The source address of the CSNP should be the Area Proxy System Id.

A Level 2 PSNP sent to an Outside Router MUST NOT contain any information about an LSP with a system identifier found in the Level 1 LSDB. If an Inside Edge Router filters a PSNP and there

is no remaining content, then the PSNP MUST NOT be sent. The source address of the PSNP should be the Area Proxy System Id.

6. Acknowledgments

The authors would like to thank Bruno Decraene and Gunter Van De Velde for their many helpful comments. The authors would also like to thank a small group that wishes to remain anonymous for their valuable contributions.

7. IANA Considerations

This memo requests that IANA allocate and assign one code point from the IS-IS TLV Codepoints registry for the Area Pseudonode TLV (XXX).

IANA is also requested to allocate and assign one code point from the IS-IS Router Capability TLV sub-TLV registry for the Area Proxy Capability (YYY).

8. Security Considerations

This document introduces no new security issues. Security of routing within a domain is already addressed as part of the routing protocols themselves. This document proposes no changes to those security architectures.

9. References

9.1. Normative References

- [I-D.ietf-lsr-dynamic-flooding]
Li, T., Psenak, P., Ginsberg, L., Chen, H., Przygienda, T., Cooper, D., Jalil, L., and S. Dontula, "Dynamic Flooding on Dense Graphs", draft-ietf-lsr-dynamic-flooding-03 (work in progress), June 2019.
- [ISO10589]
International Organization for Standardization, "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)", ISO/IEC 10589:2002, 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>>.

- [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>>.
- [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>>.
- [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>>.

9.2. Informative References

- [Clos] Clos, C., "A Study of Non-Blocking Switching Networks", The Bell System Technical Journal Vol. 32(2), DOI 10.1002/j.1538-7305.1953.tb01433.x, March 1953, <<http://dx.doi.org/10.1002/j.1538-7305.1953.tb01433.x>>.

9.3. URIs

- [1] <https://tools.ietf.org/html/bcp14>

Authors' Addresses

Tony Li
Arista Networks
5453 Great America Parkway
Santa Clara, California 95054
USA

Email: tony.li@tony.li

Sarah Chen
Arista Networks
5453 Great America Parkway
Santa Clara, California 95054
USA

Email: sarahchen@arista.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 10, 2020

A. Przygienda
C. Bowers
Juniper
Y. Lee
A. Sharma
Comcast
R. White
Juniper
January 7, 2020

IS-IS Flood Reflection
draft-przygienda-lsr-flood-reflection-01

Abstract

This document describes an optional ISIS extension that allows the creation of IS-IS flood reflection topologies. Flood reflection allows the creation of topologies where L1 areas provide transit forwarding for L2 destinations within an L2 topology. It accomplishes this by creating L2 flood reflection adjacencies within each L1 area. The L2 flood reflection adjacencies are used to flood L2 LSPDUs, and they are used in the L2 SPF computation. However, they are not used for forwarding. This arrangement gives the L2 topology better scaling properties. In addition, only those routers directly participating in flood reflection have to support the feature. This allows for the incremental deployment of scalable L1 transit areas in an existing network, without the necessity of upgrading other routers in the network.

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

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

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 10, 2020.

Copyright Notice

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

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

Table of Contents

1. Description	2
2. Further Details	8
3. Flood Reflection TLV	8
4. Flood Reflection Discovery Sub-TLV	10
5. Flood Reflection Adjacency Sub-TLV	10
6. Flood Reflection Discovery	11
7. Flood Reflection Adjacency Formation	12
8. Redistribution of Prefixes	12
9. Route Computation	13
10. Special Considerations	13
11. IANA Considerations	14
11.1. New IS-IS TLV Codepoint	14
11.2. Sub TLVs for TLV 242	14
11.3. Sub TLVs for TLV 22, 23, 25, 141, 222, and 223	15
12. Security Considerations	15
13. Acknowledgements	15
14. References	15
14.1. Informative References	15
14.2. Normative References	15
Authors' Addresses	16

1. Description

Due to the inherent properties of link-state protocols the number of IS-IS routers within a flooding domain is limited by processing and flooding overhead on each node. While that number can be maximized

by well written implementations and techniques such as exponential back-offs, IS-IS will still reach a saturation point where no further routers can be added to a single flooding domain. In some L2 backbone deployment scenarios, this limit presents a significant challenge.

The traditional approach to increasing the scale of an IS-IS deployment is to break it up into multiple L1 flooding domains and a single L2 backbone. This works well for designs where an L2 backbone connects L1 access topologies, but it is limiting where a large L2 is supposed to span large number of routers. In such scenarios, an alternative approach is to consider multiple L2 flooding domains connected together via L1 flooding domains. In other words, L2 flooding domains are connected by "L1/L2 lanes" through the L1 areas to form a single L2 backbone again. Unfortunately, in its simplest implementation, this requires the inclusion of most, or all, of the transit L1 routers as L1/L2 to allow traffic to flow along optimal paths through such transit areas. Consequently, this approach fails to reduce the number of L2 routers involved, so it fails to increase the scalability of the L2 backbone.

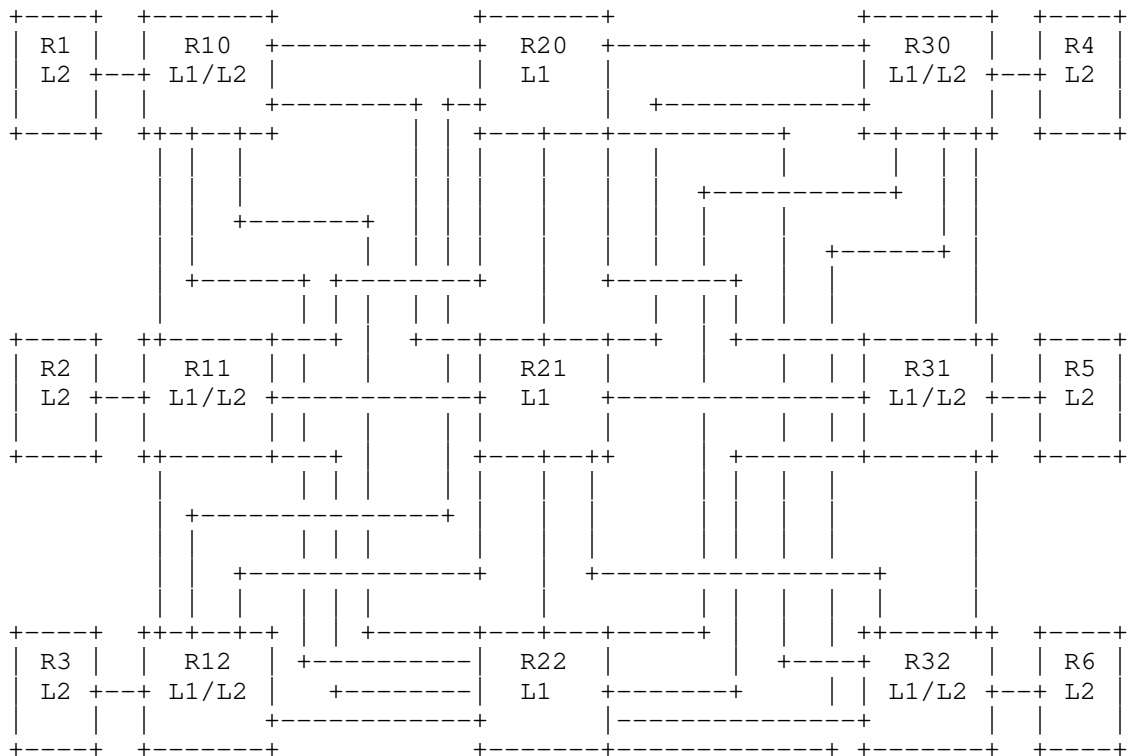


Figure 1: Example topology

Figure 1 is an example of a network where a topologically rich L1 area is used to provide transit between six different L2-only routers (R1-R6). Note that the six L2-only routers do not have connectivity to one another over L2 links. To take advantage of the abundance of paths in the L1 transit area, all the intermediate systems could be placed into both L1 and L2, but this essentially combines the separate L2 flooding domains into a single one, triggering again maximum L2 scale limitation we try to address in first place.

A more effective solution would allow to reduce the number of links and routers exposed in L2, while still utilizing the full L1 topology when forwarding through the network.

[RFC8099] describes Topology Transparent Zones (TTZ) for OSPF. The TTZ mechanism represents a group of OSPF routers as a full mesh of adjacencies between the routers at the edge of the group. A similar mechanism could be applied to ISIS as well. However, a full mesh of adjacencies between edge routers (or L1/L2 nodes) significantly

limits the scale of the topology. The topology in Figure 1 has 6 L1/L2 nodes. Figure 2 illustrates a full mesh of L2 adjacencies between the 6 L1/L2 nodes, resulting in $(5 * 6)/2 = 15$ L2 adjacencies. In a somewhat larger topology containing 20 L1/L2 nodes, the number of L2 adjacencies in a full mesh rises to 190.

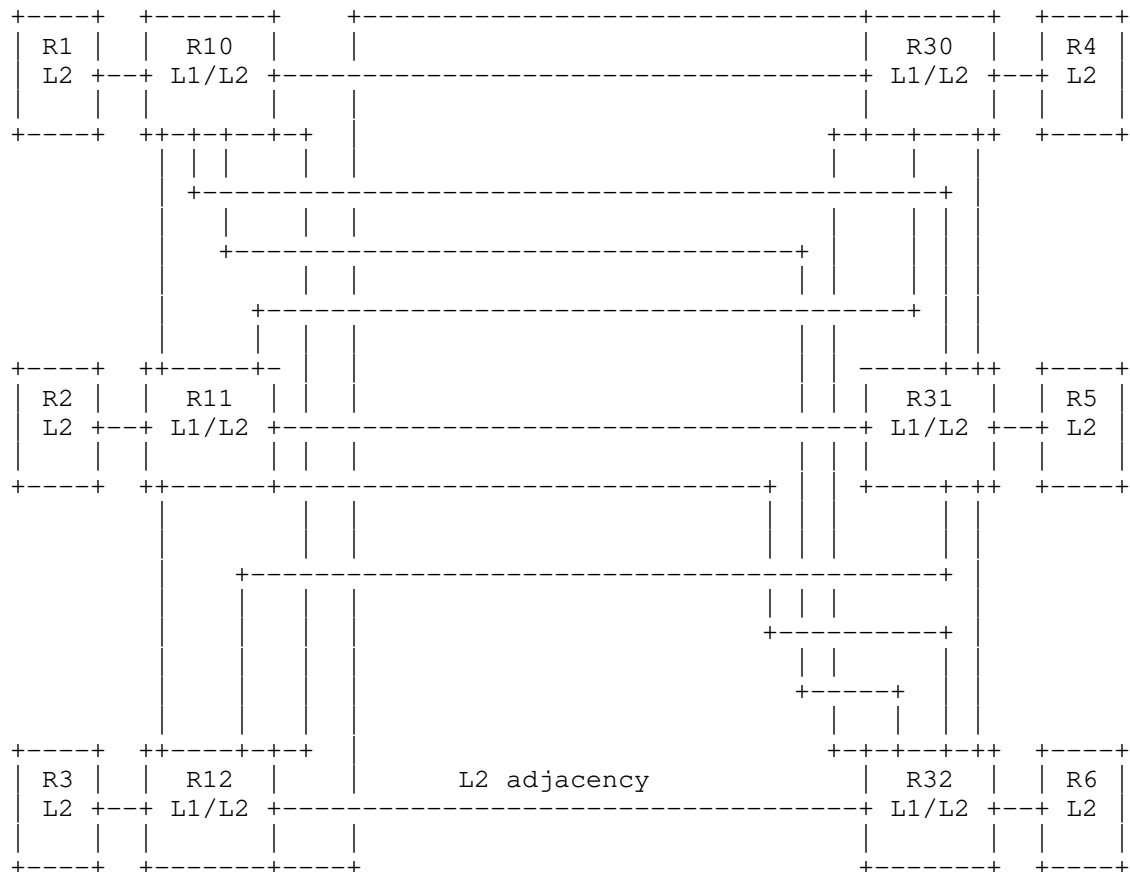


Figure 2: Example topology represented in L2 with a full mesh of L2 adjacencies between L1/L2 nodes

BGP, as specified in [RFC4271], faced a similar scaling problem, which has been solved in many networks by deploying BGP route reflectors [RFC4456]. We note that BGP route reflectors do not necessarily have to be in the forwarding path of the traffic. This incongruity of forwarding and control path for BGP route reflectors

allows the control plane to scale independently of the forwarding plane.

We propose here a similar solution for IS-IS. A simple example of what a flood reflector control plane approach would look like is shown in Figure 3, where router R21 plays the role of a flood reflector. Each L1/L2 ingress/egress router builds a tunnel to the flood reflector, and an L2 adjacency is built over each tunnel. In this solution, we need only 6 L2 adjacencies, instead of the 15 needed for a full mesh. In a somewhat larger topology containing 20 L1/L2 nodes, this solution requires only 20 L2 adjacencies, instead of the 190 need for a full mesh. Multiple flood reflectors can be used, allowing the network operator to balance between resilience, path utilization, and state in the control plane. The resulting L2 adjacency scale is $R \cdot n$, where R is the number of flood reflectors used and n is the number of L1/L2 nodes. This compares quite favorably with $n \cdot (n-1)/2$ L2 adjacencies required in a fully meshed L2 solution.

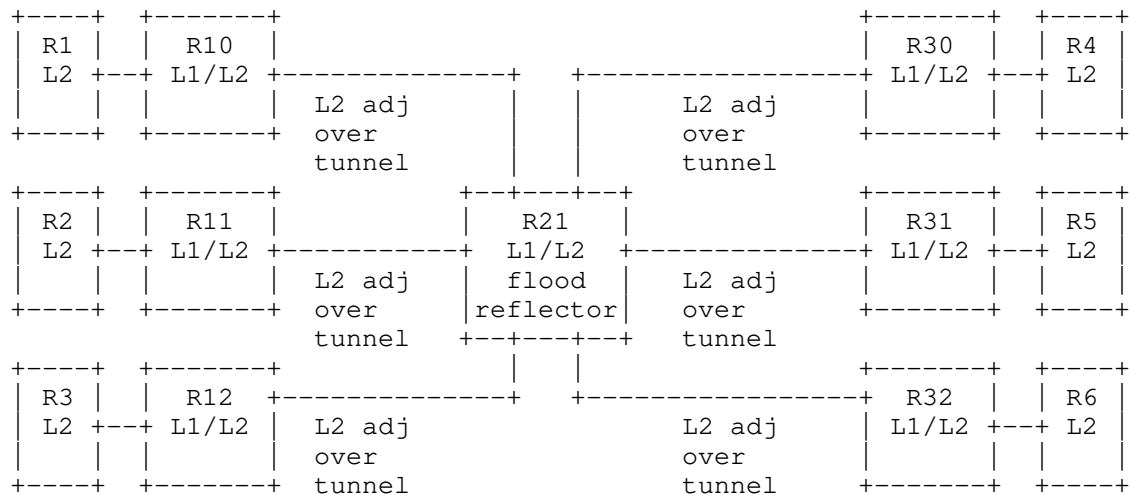


Figure 3: Example topology represented in L2 with L2 adjacencies from each L1/L2 node to a single flood reflector

As illustrated in Figure 3, when R21 plays the role of flood reflector, it provides L2 connectivity among all of the previously disconnected L2 islands by refloding all L2 LSPDUs. At the same time, R20 and R22 remain L1-only routers. L1-only routers and L1-only links are not visible in L2. In this manner, the flood

reflector allows us provide L2 control plane connectivity in a scalable manner.

As described so far, the solution illustrated in Figure 3 relies only on currently standardized ISIS functionality. Without new functionality, however, the data traffic will traverse only R21. This will unnecessarily create a bottleneck at R21 since there is still available capacity in the paths crossing the L1-only routers R20 and R22.

Hence, some new functionality is necessary to allow the L1/L2 edge nodes (R10-12 and R30-32 in Figure 3) to recognize that the L2 adjacency to R21 should not be used for forwarding. The L1/L2 edge nodes should forward traffic that would normally be forwarded over the L2 adjacency to R21 over L1 links instead. This would allow the forwarding within the L1 area to use the L1-only nodes and links shown in Figure 1 as well. It allows networks to be built that use the entire forwarding capacity of the L1 areas, while at the same time introducing control plane scaling benefits provided by L2 flood reflectors.

This document defines all extensions necessary to support flood reflector deployment:

- o A 'flood reflector adjacency' for all the adjacencies built for the purpose of reflecting flooding information. This allows these 'flood reflectors' to participate in the IS-IS control plane without being used in the forwarding plane. This is a purely local operation on the L1/L2 ingress; it does not require replacing or modifying any routers not involved in the reflection process. Deployment-wise, it is far less tricky to just upgrade the routers involved in flood reflection rather than have a flag day on the whole ISIS domain.
- o A full mesh of L1 tunnels between the L1/L2 routers, ideally load-balancing across all available L1 links. This harnesses all forwarding paths between the L1/L2 edge nodes without injecting unneeded state into the L2 flooding domain or creating 'choke points' at the 'flood reflectors' themselves. A solution without tunnels is also possible by judicious scoping of reachability information between the levels.
- o Some way to support reflector redundancy, and potentially some way to auto-discover and advertise such adjacencies as flood reflector adjacencies. Such advertisements may allow L2 nodes outside the L1 to perform optimizations in the future based on this information.

2. Further Details

Several considerations should be noted in relation to such a flood reflection mechanism.

First, this allows multi-area IS-IS deployments to scale without any major modifications in the IS-IS implementation on most of the nodes deployed in the network. Unmodified (traditional) L2 routers will compute reachability across the transit L1 area using the flood reflector adjacencies.

Second, the flood reflectors are not required to participate in forwarding traffic through the L1 transit area. These flood reflectors can be hosted on virtual devices outside the forwarding topology.

Third, astute readers will realize that flooding reflection may cause the use of suboptimal paths. This is similar to the BGP route reflection suboptimal routing problem described in [ID.draft-ietf-idr-bgp-optimal-route-reflection-19]. The L2 computation determines the egress L1/L2 and with that can create illusions of ECMP where there is none. And in certain scenarios lead to an L1/L2 egress which is not globally optimal. This represents a straightforward instance of the trade-off between the amount of control plane state and the optimal use of paths through the network often encountered when aggregating routing information.

One possible solution to this problem is to expose additional topology information into the L2 flooding domains. In the example network given, links from router 01 to router 02 can be exposed into L2 even when 01 and 02 are participating in flood reflection. This information would allow the L2 nodes to build 'shortcuts' when the L2 flood reflected part of the topology looks more expensive to cross distance wise.

Another possible variation is for an implementation to approximate with the L1 tunnel cost the cost of the underlying topology.

Redundancy can be achieved by building multiple flood reflectors in the L1 area. Multiple flood reflectors do not need any synchronization mechanisms amongst themselves, except standard ISIS flooding and database maintenance procedures.

3. Flood Reflection TLV

The Flood Reflection TLV is a new top-level TLV that MAY appear in IIHs. The Flood Reflection TLV indicates the flood reflector cluster (based on Flood Reflection Cluster ID) that a given router is

configured to participate in. It also indicates whether the router is configured to play the role of either flood reflector or flood reflector client. The Flood Reflection Cluster ID and flood reflector roles advertised in the IIHs are used to ensure that flood reflector adjacencies are only formed between a flood reflector and flood reflector client, and that the Flood Reflection Cluster IDs match. The Flood Reflection 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      | C |  RESERVED  |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Flood Reflection Cluster ID
+-----+-----+-----+-----+-----+-----+-----+-----+
|      Sub-TLVs ...
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Type: TBD

Length: The length, in octets, of the following fields.

C (Client): This bit is set to indicate that the router acts as a flood reflector client. When this bit is NOT set, the router acts as a flood reflector. On a given router, the same value of the C-bit MUST be advertised across all interfaces advertising the Flood Reflection TLV in IIHs.

RESERVED: This field is reserved for future use. It MUST be set to 0 when sent and MUST be ignored when received.

Flood Reflection Cluster ID: Flood Reflection Cluster Identifier. These same 32-bit value MUST be assigned to all of the flood reflectors and flood reflector clients in the L1 area. The value MUST be unique across different L1 areas within the IGP domain. On a given router, the same value of the Flood Reflection Cluster ID MUST be advertised across all interfaces advertising the Flood Reflection TLV in IIHs.

Sub-TLVs: Optional sub-TLVs. For future extensibility, the format of the Flood Reflection TLV allows for the possibility of including optional sub-TLVs. No sub-TLVs of the Flood Reflection TLV are defined in this document.

The Flood Reflection TLV MUST NOT appear more than once in an IIH. A router receiving multiple Flood Reflection TLVs in the same IIH SHOULD use the values in the first TLV.

4. Flood Reflection Discovery Sub-TLV

Flood Reflection Discovery sub-TLV is advertised as a sub-TLV of the IS-IS Router Capability TLV-242, defined in [RFC7981]. The Flood Reflection Discovery sub-TLV is advertised in L1 LSPs with area flooding scope in order to enable the auto-discovery of flood reflection capabilities and the automatic creation of L2 tunnels to be used as flood reflector adjacencies. The Flood Reflection Discovery 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      |C|   Reserved   |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Flood Reflection Cluster ID
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Type: TBD

Length: The length, in octets, of the following fields.

C (Client): This bit is set to indicate that the router acts as a flood reflector client. When this bit is NOT set, the router acts as a flood reflector.

RESERVED: This field is reserved for future use. It MUST be set to 0 when sent and MUST be ignored when received.

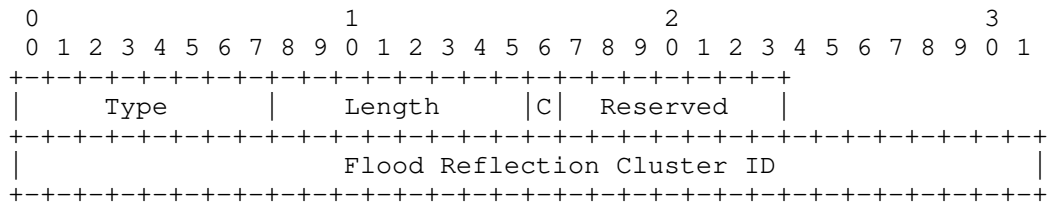
Flood Reflection Cluster ID: The Flood Reflection Cluster Identifier is the same as that defined in the Flood Reflection TLV.

The Flood Reflection Discovery sub-TLV MUST NOT appear more than once in TLV 242. A router receiving multiple Flood Reflection Discovery sub-TLVs in TLV 242 SHOULD use the values in the first sub-TLV.

5. Flood Reflection Adjacency Sub-TLV

The Flood Reflection Adjacency sub-TLV is advertised as a sub-TLV of TLVs 22, 23, 25, 141, 222, and 223. Its presence indicates that a given adjacency is a flood reflector adjacency. It is included in L2

area scope flooded LSPs. Flood Reflection Adjacency sub-TLV has the following format:



Type: TBD

Length: The length, in octets, of the following fields.

C (Client): This bit is set to indicate that the router advertising this adjacency is a flood reflector client. When this bit is NOT set, the router advertising this adjacency is a flood reflector.

RESERVED: This field is reserved for future use. It MUST be set to 0 when sent and MUST be ignored when received.

Flood Reflection Cluster ID: The Flood Reflection Cluster Identifier is the same as that defined in the Flood Reflection TLV.

The Flood Reflection Adjacency sub-TLV MUST NOT appear more than once in a given TLV. A router receiving multiple Flood Reflection Adjacency sub-TLVs in a TLV SHOULD use the values in the first sub-TLV.

6. Flood Reflection Discovery

A router participating in flood reflection MUST be configured as an L1/L2 router. It originates the Flood Reflection Discovery sub-TLV with area flooding scope in L1 only. Normally, all routers on the edge of the L1 area (those having traditional L2 adjacencies) will advertise themselves as route reflector clients. Therefore, a flood reflector client will have both traditional L2 adjacencies and flood reflector L2 adjacencies.

A router acting as a flood reflector MUST NOT have any traditional L2 adjacencies. It will be an L1/L2 router only by virtue of having flood reflector L2 adjacencies. A router desiring to act as a flood reflector will advertise itself as such using the Flood Reflection Discovery sub-TLV in L1.

A given flood reflector or flood reflector client can only participate in a single cluster, as determined by the value of its Flood Reflection Cluster ID.

Upon reception of Flood Reflection Discovery sub-TLVs, a router acting as flood reflector client MUST initiate a tunnel towards each flood reflector with which it shares an Flood Reflection Cluster ID. The L2 adjacencies formed over such tunnels MUST be marked as flood reflector adjacencies. If the client has a direct L2 adjacency with the flood reflector it SHOULD use it instead of instantiating a new tunnel.

Upon reception of Flood Reflection Discover TLVs, a router acting as a flood reflector client MAY initiate tunnels with L1-only adjacencies towards all the other flood reflector clients in its cluster. These tunnels MAY be used for forwarding to improve the load-balancing characteristics of the L1 area.

7. Flood Reflection Adjacency Formation

In order to simplify both implementations and network deployments, we do not allow the formation of complex hierarchies of flood reflectors and clients. All flood reflectors and flood reflector clients in the same L1 area MUST share the same Flood Reflector Cluster ID. A flood reflector MUST only form flood reflection adjacencies with flood reflector clients. A flood reflector MUST NOT form any traditional L2 adjacencies. Flood reflector clients MUST only form flood reflection adjacencies with flood reflectors. Flood reflector clients may form traditional L2 adjacencies with flood reflector clients or nodes not participating in flood reflection.

The Flood Reflector Cluster ID and flood reflector roles advertised in the Flood Reflection TLVs in IIHs are used to ensure that flood reflection adjacencies that are established meet the above criteria.

Once a flood reflection adjacency is established, the flood reflector and the flood reflector client MUST advertise the adjacency by including the Flood Reflection Adjacency Sub-TLV in the Extended IS reachability TLV or MT-ISN TLV.

8. Redistribution of Prefixes

In some scenarios, L2 prefixes need to be redistributed into L1 by the route reflector clients. However, if a L1 area edge router doesn't have any L2 flood reflector adjacencies, then it cannot be the shortest path egress in the L2 topology. Therefore, flood reflector client SHOULD only redistribute L2 prefixes into L1 if it has an L2 flood reflector adjacency. The L2 prefixes advertisements

redistributed into L1 SHOULD be normally limited to L2 intra-area routes (as defined in [RFC7775]), if the information exists to distinguish them from other L2 prefix advertisements.

On the other hand, in topologies that make use of flood reflection to hide the structure of L1 areas while still providing transit forwarding across them, we generally do not need to redistribute L1 prefixes advertisements into L2.

In deployment scenarios where L1 tunnels are not used, all L1/L2 edge nodes MUST be flood reflector clients.

9. Route Computation

To ensure loop-free routing, the route reflection client MUST follow the normal L2 computation to determine L2 routes. This is because nodes outside the L1 area will generally not be aware that flood reflection is being performed. The flood reflection clients need to produce the same result for the L2 route computation as a router not participating in flood reflection. However, a flood reflector client will not necessarily use a given L2 route for forwarding. For an L2 route that uses a flood reflection adjacency as a next-hop, the flood reflection client may use the next-hop from an L1 route instead.

On the reflection client, after L2 and L1 computation, all flood reflector adjacencies used as next-hops for L2 routes MUST be examined and replaced with the correct L1 tunnel next-hop to the egress. Alternatively, if the ingress has adequate reachability information to ensure forwarding towards destination via L1 routes, L2 routes using flood reflector adjacencies as next-hops can be omitted entirely. Due to the rules in Section 7 the computation in the resulting topology is relatively simple, the L2 SPF from a flood reflector client is guaranteed to reach within a hop the Flood Reflector and in the following hop the L2 egress to which it has a L1 forwarding tunnel. However, if the topology has L2 paths which are not route reflected and look "shorter" than the path through the Flood Reflector then the computation will have to track the egress out of the L1 domain by a more advanced algorithm.

10. Special Considerations

In pathological cases setting the overload bit in L1 (but not in L2) can partition L1 forwarding, while allowing L2 reachability through flood reflector adjacencies to exist. In such a case a node cannot replace a route through a flood reflector adjacency with a L1 shortcut and the client can use the L2 tunnel to the flood reflector for forwarding while it MUST initiate an alarm and declare misconfiguration.

A flood reflector with directly L2 attached prefixes should advertise those in L1 as well since based on preference of L1 routes the clients will not try to use the L2 flood reflector adjacency to route the packet towards them. A very, very corner case is when the flood reflector is reachable via L2 flood reflector adjacency (due to underlying L1 partition) only in which case the client can use the L2 tunnel to the flood reflector for forwarding towards those prefixes while it MUST initiate an alarm and declare misconfiguration.

Instead of modifying the computation procedures one could imagine a flood reflector solution where the Flood Reflector would re-advertise the L2 prefixes with a 'third-party' next-hop but that would have less desirable convergence properties than the solution proposed and force a fork-lift of all L2 routers to make sure they disregard such prefixes unless in the same L1 domain as the Flood Reflector.

Depending on pseudo-node choice in case of a broadcast domain with multiple flood reflectors attached this can lead to a partitioned LAN and hence a router discovering such a condition MUST initiate an alarm and declare misconfiguration.

11. IANA Considerations

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

11.1. New IS-IS TLV Codepoint

This document requests the following IS-IS TLV:

Value	Name	IIH	LSP	SNP	Purge
-----	-----	---	---	---	-----
TBD1	Flood Reflection	y	n	n	n

11.2. Sub TLVs for TLV 242

This document request the following registration in the "sub-TLVs for TLV 242" registry.

Type	Description
----	-----
TBD2	Flood Reflection Discovery

11.3. Sub TLVs for TLV 22, 23, 25, 141, 222, and 223

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

Type	Description	22	23	25	141	222	223
----	-----	---	---	---	---	---	---
TBD3	Flood Reflector Adjacency	y	y	y(s)	y	y	y

12. Security Considerations

This document introduces no new security concerns to ISIS or other specifications referenced in this document.

13. Acknowledgements

The authors thank Shraddha Hegde, Peter Psenak, and Les Ginsberg for their thorough review and detailed discussions.

14. References

14.1. Informative References

- [ID.draft-ietf-idr-bgp-optimal-route-reflection-19]
Raszuk et al., R., "BGP Optimal Route Reflection", July 2019.
- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <<https://www.rfc-editor.org/info/rfc4271>>.
- [RFC4456] Bates, T., Chen, E., and R. Chandra, "BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)", RFC 4456, DOI 10.17487/RFC4456, April 2006, <<https://www.rfc-editor.org/info/rfc4456>>.
- [RFC8099] Chen, H., Li, R., Retana, A., Yang, Y., and Z. Liu, "OSPF Topology-Transparent Zone", RFC 8099, DOI 10.17487/RFC8099, February 2017, <<https://www.rfc-editor.org/info/rfc8099>>.

14.2. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC7775] Ginsberg, L., Litkowski, S., and S. Previdi, "IS-IS Route Preference for Extended IP and IPv6 Reachability", RFC 7775, DOI 10.17487/RFC7775, February 2016, <<https://www.rfc-editor.org/info/rfc7775>>.
- [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>>.

Authors' Addresses

Tony Przygienda
Juniper
1137 Innovation Way

Sunnyvale, CA

USA

Email: prz@juniper.net

Chris Bowers
Juniper
1137 Innovation Way

Sunnyvale, CA

USA

Email: cbowers@juniper.net

Yiu Lee
Comcast
1800 Bishops Gate Blvd
Mount Laurel, NJ 08054
US

Email: Yiu_Lee@comcast.com

Alankar Sharma
Comcast
1800 Bishops Gate Blvd
Mount Laurel, NJ 08054
US

Email: Alankar_Sharma@comcast.com

Russ White
Juniper
1137 Innovation Way

Sunnyvale, CA

USA

Email: russw@juniper.net

LSR Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 18, 2022

A. Wang
China Telecom
G. Mishra
Verizon Inc.
Z. Hu
Y. Xiao
Huawei Technologies
October 15, 2021

Prefix Unreachable Announcement
draft-wang-lsr-prefix-unreachable-announcement-08

Abstract

This document describes a mechanism to solve an existing issue with Longest Prefix Match (LPM), that exists where an operator domain is divided into multiple areas or levels where summarization is utilized. This draft addresses a fail-over issue related to a multi areas or levels domain, where a link or node down event occurs resulting in an LPM component prefix being omitted from the FIB resulting in black hole sink of routing and connectivity loss. This draft introduces a new control plane convergence signaling mechanism using a negative prefix called Prefix Unreachable Announcement Mechanism(PUAM), utilized to detect a link or node down event and signal the RIB that the event has occurred to force immediate control plane convergence.

Status of This Memo

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

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

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

This Internet-Draft will expire on April 18, 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
2. Conventions used in this document	3
3. Scenario Description	3
3.1. Inter-Area Node Failure Scenario	4
3.2. Inter-Area Links Failure Scenario	4
4. PUA (Prefix Unreachable Advertisement) Procedures	5
5. MPLS and SRv6 LPM based BGP Next-hop Failure Application	5
6. PUAM Capabilities Announcement	6
7. Implementation Consideration	7
8. Deployment Considerations	7
9. Security Considerations	8
10. IANA Considerations	8
11. Acknowledgement	9
12. Normative References	9
Authors' Addresses	10

1. Introduction

As part of an operator optimized design criteria, a critical requirement is to limit Shortest Path First (SPF) churn which occurs within a single OSPF area or ISIS level. This is accomplished by sub-dividing the IGP domain into multiple areas for flood reduction of intra area prefixes so they are contained within each discrete area to avoid domain wide flooding.

OSPF and ISIS have a default and summary route mechanism which is performed on the OSPF area border router or ISIS L1-L2 node. The OSPF summary route is triggered to be advertised conditionally when at least one component prefix exists within the non-zero area. ISIS Level-L1-L2 node as well generate a summary prefix into the level-2 backbone area for Level 1 area prefixes that is triggered to be

advertised conditionally when at least a single component prefix exists within the Level-1 area. ISIS L1-L2 node with attach bit set also generates a default route into each Level-1 area along with summary prefixes generated for other Level-1 areas.

Operators have historically relied on MPLS architecture which is based on exact match host route FEC binding for single area. [RFC5283] LDP inter-area extension provides the ability to LPM, so now the RIB match can now be a summary match and not an exact match of a host route of the egress PE for an inter-area LSP to be instantiated. SRV6 routing framework utilizes the IPv6 data plane standard IGP LPM. When operators start to migrate from MPLS LSP based host route bootstrapped FEC binding, to SRV6 routing framework, the IGP LPM now comes into play with summarization which will influence the forwarding of traffic when a link or node event occurs for a component prefix within the summary range resulting in black hole routing of traffic.

The motivation behind this draft is based on either MPLS LPM FEC binding, or SRv6 BGP service overlay using traditional unicast routing (uRIB) LPM forwarding plane where the IGP domain has been carved up into OSPF or ISIS areas and summarization is utilized. In this scenario where a failure conditions result in a black hole of traffic where multiple ABRs exist and either the area is partitioned or other link or node failures occur resulting in the component prefix host route missing within the summary range. Summarization of inter-area types routes propagated into the backbone area for flood reduction are made up of component prefixes. It is these component prefixes that the PUAM tracks to ensure traffic is not black hole sink routed due to a PE or ABR failure. The PUA mechanism ensures immediate control plane convergence with ABR or PE node switchover when area is partitioned or ABR has services down to avoid black hole of traffic.

2. Conventions used in this document

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

3. Scenario Description

Figure 1 illustrates the topology scenario when OSPF or ISIS is running in multi areas or multi levels domain. R0-R4 are routers in backbone area, S1-S4,T1-T4 are internal routers in area 1 and area 2 respectively. R1 and R3 are area border routers or ISIS Level 1-2 border nodes between area 0 and area 1. R2 and R4 are area border routers between area 0 and area 2.

S1/S4 and T2/T4 PEs peer to customer CEs for overlay VPNs. Ps1/Ps4 is the loopback0 address of S1/S4 and Pt2/Pt4 is the loopback0 address of T2/T4.

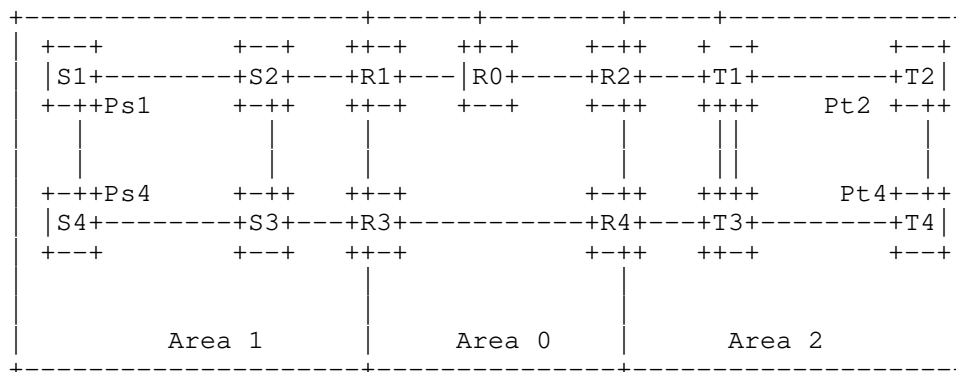


Figure 1: OSPF Inter-Area Prefix Unreachable Announcement Scenario

3.1. Inter-Area Node Failure Scenario

If the area border router R2/R4 does the summary action, then one summary address that cover the prefixes of area 2 will be announced to area 0 and area 1, instead of the detail address. When the node T2 is down, Pt2 bgp next hop becomes unreachable while the LPM summary prefix continues to be advertised into the backbone area. Except the border router R2/R4, the other routers within area 0 and area 1 do not know the unreachable status of the Pt2 bgp next hop prefix. Traffic will continue to forward LPM match to prefix Pt2 and will be dropped on the ABR or Level 1-2 border node resulting in black hole routing and connectivity loss. Customer overlay VPN dual homed to both S1/S4 and T2/R4, traffic will not be able to fail-over to alternate egress PE T4 bgp next hop Pt4 due to the summarization.

3.2. Inter-Area Links Failure Scenario

In a link failure scenario, if the link between T1/T2 and T1/T3 are down, R2 will not be able to reach node T2. But as R2 and R4 do the summary announcement, and the summary address covers the bgp next hop prefix of Pt2, other nodes in area 0 area 1 will still send traffic to T2 bgp next hop prefix Pt2 via the border router R2, thus black hole sink routing the traffic.

In such a situation, the border router R2 should notify other routers that it can't reach the prefix Pt2, and lets the other ABRs(R4) that can reach prefix Pt2 advertise one specific route to Pt2, then the

internal routers will select R4 as the bypass router to reach prefix Pt2.

4. PUA (Prefix Unreachable Advertisement) Procedures

[RFC7794] and [I-D.ietf-lsr-ospf-prefix-originator] draft both define one sub-tlv to announce the originator information of the one prefix from a specified node. This draft utilizes such TLV for both OSPF and ISIS to signal the negative prefix in the perspective PUAM when a link or node goes down.

ABR detects link or node down and floods PUAM negative prefix advertisement along with the summary advertisement according to the prefix-originator specification. The ABR or ISIS L1-L2 border node has the responsibility to add the prefix originator information when it receives the Router LSA from other routers in the same area or level.

When the ABR or ISIS L1-L2 border node generates the summary advertisement based on component prefixes, the ABR will announce one new summary LSA or LSP which includes the information about this down prefix, with the prefix originator set to NULL. The number of PUAMs is equivalent to the number of links down or nodes down. The LSA or LSP will be propagated with standard flooding procedures.

If the nodes in the area receive the PUAM flood from all of its ABR routers, they will start BGP convergence process if there exist BGP session on this PUAM prefix. The PUAM creates a forced fail over action to initiate immediate control plane convergence switchover to alternate egress PE. Without the PUAM forced convergence the down prefix will yield black hole routing resulting in loss of connectivity.

When only some of the ABRs can't reach the failure node/link, as that described in Section 3.2, the ABR that can reach the PUAM prefix should advertise one specific route to this PUAM prefix. The internal routers within another area can then bypass the ABRs that can't reach the PUAM prefix, to reach the PUAM prefix.

5. MPLS and SRv6 LPM based BGP Next-hop Failure Application

In an MPLS or SR-MPLS service provider core, scalability has been a concern for operators which have split up the IGP domain into multiple areas to avoid SPF churn. Normally, MPLS FEC binding for LSP instantiation is based on egress PE exact match of a host route Looback0. [RFC5283] LDP inter-area extension provides the ability to LPM, so now the RIB match can now be a summary match and not an exact match of host route of the egress PE for an inter-area LSP to be

instantiated. The caveat related to this feature that has prevented operators from using the [RFC5283] LDP inter-area extension concept is that when the component prefixes are now hidden in the summary prefix, and thus the visibility of the BGP next-hop attribute is lost.

In a case where a PE is down, and the [RFC5283] LDP inter-area extension LPM summary is used to build the LSP inter-area, the LSP remains partially established black hole on the ABR performing the summarization. This major gap with [RFC5283] inter-area extension forces operators into a workaround of having to flood the BGP next-hop domain wide. In a small network this is fine, however if you have 1000s PEs and many areas, the domain wide flooding can be painful for operators as far as resource usage memory consumption and computational requirements for RIB / FIB / LFIB label binding control plane state. The ramifications of domain wide flooding of host routes is described in detail in [RFC5302] domain wide prefix distribution with 2 level ISIS Section 1.2 - Scalability. As SRv6 utilizes LPM, this problem exists as well with SRv6 when IGP domain is broken up into areas and summarization is utilized.

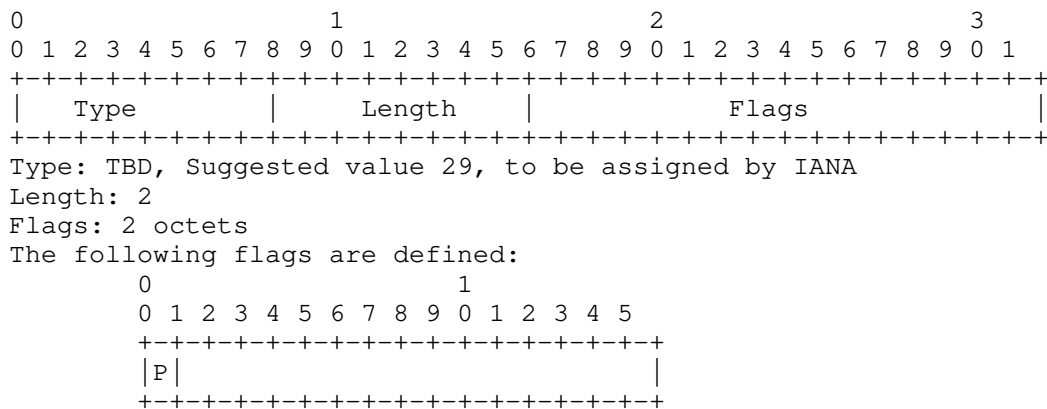
PUAM is now able to provide the negative prefix component flooded across the backbone to the other areas along with the summary prefix, which is now immediately programmed into the RIB control plane. MPLS LSP exact match or SRv6 LPM match over fail over path can now be established to the alternate egress PE. No disruption in traffic or loss of connectivity results from PUAM. Further optimizations such as LFA and BFD can be done to make the data plane convergence hitless. The PUAM solution applies to MPLS or SR-MPLS where LDP inter-area extension is utilized for LPM aggregate FEC, as well a SRv6 IPv6 control plane LPM match summarization of BGP next hop.

6. PUAM Capabilities Announcement

When not all of the nodes in one area support the PUAM information, there are possibilities to form traffic loop. To avoid this happen, the ABR should not send PUAM information to one area until it ensures that all of nodes in this area can parse the PUAM information. To accomplish this, this draft defines the capabilities sub-TLV as the followings:

For OSPFv2, this bit (Bit number TBD, suggest bit 6, 0x20) should be carried in "OSPF Router-LSA Option", as that described in [RFC2328]. For OSPFv3, one bit (Bit number TBD, suggest bit 8) should be defined to indicate the router's capabilities to support PUAM that described in this draft, the defined bit should be carried in "OSPF Router Informational Capabilities" TLV, which is described in [RFC7770]. For ISIS, one new sub-TLV(Type TBD, suggest 29), PUAM Capabilities

sub-TLV, which is included in the "IS-IS Router CAPABILITY TLV" [RFC7981] is defined in the followings:



where:

P-flag: If set, the router supports PUA information.

Figure 2: PUA Capabilities sub-TLV format

7. Implementation Consideration

Considering the balances of reachable information and unreachable information announcement capabilities, the implementation of this mechanism should set one MAX_Address_Announcement (MAA) threshold value that can be configurable. Then, the ABR should make the following decisions to announce the prefixes:

1. If the number of unreachable prefixes is less than MAA, the ABR should advertise the summary address and the PUAM.
2. If the number of reachable address is less than MAA, the ABR should advertise the detail reachable address only.
3. If the number of reachable prefixes and unreachable prefixes exceed MAA, then advertise the summary address with MAX metric.

8. Deployment Considerations

To support the PUAM advertisement, the ABRs should be upgraded according to the procedures described in Section 4. The PEs that want to accomplish the BGP switchover that described in Section 3.1 and Section 5 should also be upgraded to act upon the receive of the PUAM message. Other nodes within the network can ignore such PUAM message if they don't care or don't support.

As described in Section 4, the ABR will advertise the PUAM message once it detects there is link or node down within the summary address. In order to reduce the unnecessary advertisements of PUAM messages on ABRs, the ABRs should support the configuration of the protected prefixes. Based on such information, the ABR will only advertise the PUAM message when the protected prefixes (for example, the loopback addresses of PEs that run BGP) that within the summary address is missing.

The advertisement of PUAM message should only last one configurable period to allow the services that run on the failure prefixes are converged or switchover. If one prefix is missed before the PUAM takes effect, the ABR will not declare its absence via the PUAM.

9. Security Considerations

Advertisement of PUAM information follow the same procedure of traditional LSA. The action based on the PUAM is clearly defined in this document for ABR or Level1/2 router and the receiver that run BGP.

There is no changes to the forward behavior of other internal routers.

10. IANA Considerations

IANA is requested to register the following in the "OSPF Router Properties Registry" and "OSPF Router Informational Capability Bits Registry" respectively.

Bit Number	Capability Name	Reference
TBD(0x20)	OSPF PUA Support	this document

Table 1: P-Bit in OSPF Router-LSA Option

Bit Number	Capability Name	Reference
TBD(bit 8)	OSPF PUA Support	this document

Table 2: OSPF Router PUA Capability Support Bit

IANA is requested to register the following in "Sub-TLVs for TLV242 (IS-IS Router CAPABILITY TLV)

Type: 29 (Suggested - to be assigned by IANA)

Description: PUA Support Capabilities

11. Acknowledgement

Thanks Peter Psenak, Les Ginsberg, Acee Lindem, Shraddha Hegde, Robert Raszuk, Tonly Li, Jeff Tantsura, Tony Przygienda and Bruno Decraene for their suggestions and comments on this draft.

12. Normative References

- [I-D.ietf-lsr-ospf-prefix-originator]
Wang, A., Lindem, A., Dong, J., Psenak, P., and K. Talaulikar, "OSPF Prefix Originator Extensions", draft-ietf-lsr-ospf-prefix-originator-12 (work in progress), April 2021.
- [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>>.
- [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <<https://www.rfc-editor.org/info/rfc2328>>.
- [RFC5283] Decraene, B., Le Roux, J.L., and I. Minei, "LDP Extension for Inter-Area Label Switched Paths (LSPs)", RFC 5283, DOI 10.17487/RFC5283, July 2008, <<https://www.rfc-editor.org/info/rfc5283>>.

- [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>>.
- [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <<https://www.rfc-editor.org/info/rfc5340>>.
- [RFC5709] Bhatia, M., Manral, V., Fanto, M., White, R., Barnes, M., Li, T., and R. Atkinson, "OSPFv2 HMAC-SHA Cryptographic Authentication", RFC 5709, DOI 10.17487/RFC5709, October 2009, <<https://www.rfc-editor.org/info/rfc5709>>.
- [RFC7770] Lindem, A., Ed., Shen, N., Vasseur, JP., Aggarwal, R., and S. Shaffer, "Extensions to OSPF for Advertising Optional Router Capabilities", RFC 7770, DOI 10.17487/RFC7770, February 2016, <<https://www.rfc-editor.org/info/rfc7770>>.
- [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>>.

Authors' Addresses

Aijun Wang
China Telecom
Beiqijia Town, Changping District
Beijing 102209
China

Email: wangaj3@chinatelecom.cn

Gyan Mishra
Verizon Inc.

Email: gyan.s.mishra@verizon.com

Zhibo Hu
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: huzhibo@huawei.com

Yaqun Xiao
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: xiaoyaqun@huawei.com