

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: March 27, 2021

A. Lindem, Ed.  
P. Psenak  
Cisco Systems  
September 23, 2020

Extensions to OSPF for Advertising Prefix Administrative Tags  
draft-acee-lsr-ospf-admin-tags-07

Abstract

It is useful for routers in an OSPFv2 or OSPFv3 routing domain to be able to associate tags with prefixes. Previously, OSPFv2 and OSPFv3 were relegated to a single tag for AS External and Not-So-Stubby-Area (NSSA) prefixes. With the flexible encodings provided by OSPFv2 Prefix/Link Attribute Advertisement and OSPFv3 Extended LSAs, multiple administrative tags may advertised for all types of prefixes. These administrative tags can be used for many applications including route redistribution policy, selective prefix prioritization, selective IP Fast-ReRoute (IPFRR) prefix protection, and many others.

The ISIS protocol supports a similar mechanism that is described in RFC 5130.

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 27, 2021.

Copyright Notice

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

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

## Table of Contents

1. Introduction . . . . .	2
1.1. Requirements Language . . . . .	3
2. 32-Bit Administrative Tag Sub-TLV . . . . .	3
3. Administrative Tag Applicability . . . . .	4
4. Protocol Operation . . . . .	5
4.1. Equal-Cost Multipath Applicability . . . . .	5
5. Security Considerations . . . . .	5
6. IANA Considerations . . . . .	6
7. Acknowledgments . . . . .	6
8. References . . . . .	6
8.1. Normative References . . . . .	6
8.2. Informative References . . . . .	7
Appendix A. 64-Bit Administrative Tag Sub-TLV . . . . .	8
Appendix B. Link Administrative Tags . . . . .	8
Authors' Addresses . . . . .	9

## 1. Introduction

It is useful for routers in an OSPFv2 [RFC2328] or OSPFv3 [RFC5340] routing domain to be able to associate tags with prefixes. Previously, OSPFv2 and OSPFv3 were relegated to a single tag for AS External and Not-So-Stubby-Area (NSSA) prefixes. With the flexible encodings provided by OSPFv2 Prefix/Link Attribute Advertisement ([RFC7684]) and OSPFv3 Extended LSA ([RFC8362]), multiple administrative tags may be advertised for all types of prefixes. These administrative tags can be used many applications including (but not limited to):

1. Controlling which routes are redistributed into other protocols for readvertisement.
2. Prioritizing selected prefixes for faster convergence and installation in the forwarding plane.
3. Identifying selected prefixes for Loop-Free Alternative (LFA) protection.

Throughout this document, OSPF is used when the text applies to both OSPFv2 and OSPFv3. OSPFv2 or OSPFv3 is used when the text is specific to one version of the OSPF protocol.

The ISIS protocol supports a similar mechanism that is described in RFC 5130 [RFC5130].

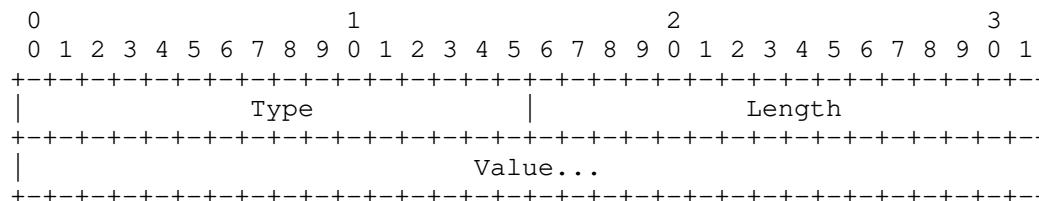
### 1.1. Requirements Language

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. 32-Bit Administrative Tag Sub-TLV

This document creates a new Administrative Tag Sub-TLV for OSPFv2 and OSPFv3. This Sub-TLV specifies one or more 32-bit unsigned integers that may be associated with an OSPF advertised prefix. The precise usage of these tags is beyond the scope of this document.

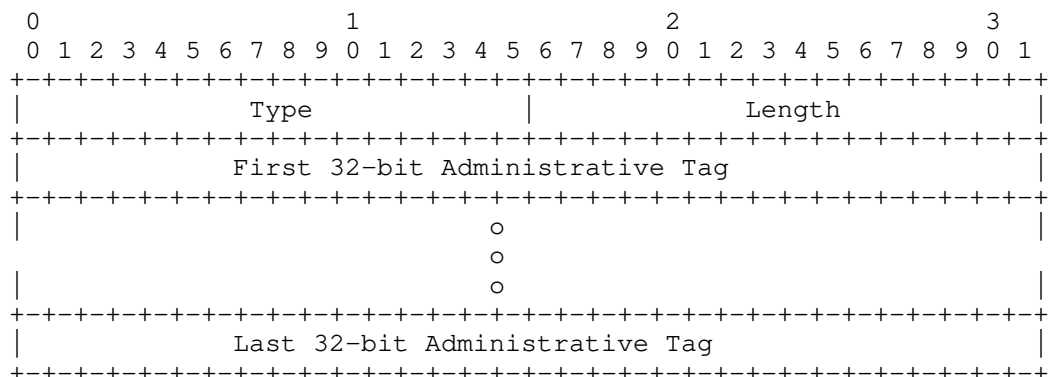
The format of this Sub-TLV is the same as the format used by the Traffic Engineering Extensions to OSPF [RFC3630]. The LSA payload consists of one or more nested Type/Length/Value (TLV) triplets. The format of each TLV is:



TLV Format

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).

The format of the 32-bit Administrative Tag TLV is as follows:



**Type** A 16-bit field set to TBD. The value MAY be different depending upon the IANA registry from which it is allocated.

**Length** A 16-bit field that indicates the length of the value portion in octets and will be a multiple of 4 octets dependent on the number of administrative tags advertised. If the sub-TLV is specified, at least one administrative tag must be advertised.

**Value** A variable length list of one or more administrative tags.

#### 32-bit Administrative Tag Sub-TLV

This sub-TLV will carry one or more 32-bit unsigned integer values that will be used as administrative tags.

### 3. Administrative Tag Applicability

The administrative tag TLV specified herein will be valid as a sub-TLV of the following TLVs specified in [RFC7684]:

1. Extended Prefix TLV advertised in the OSPFv2 Extended Prefix LSA

The administrative tag TLV specified herein will be valid as a sub-TLV of the following TLVs specified in [RFC8362]:

1. Inter-Area-Prefix TLV advertised in the E-Inter-Area-Prefix-LSA
2. Intra-Area-Prefix TLV advertised in the E-Link-LSA and the E-Intra-Area-Prefix-LSA

3. External-Prefix TLV advertised in the E-AS-External-LSA and the E-NSSA-LSA

#### 4. Protocol Operation

An OSPF router supporting this specification MUST propagate administrative tags when acting as an Area Border Router and originating summary advertisements into other areas. Similarly, an OSPF router supporting this specification and acting as an ABR for a Not-So-Stubby Area (NSSA) MUST propagate tags when translating NSSA routes to AS External advertisements [RFC3101]. The number of tags supported MAY limit the number of tags that are propagated. When propagating multiple tags, the order of the the tags must be preserved.

For configured area ranges, NSSA ranges, and configured summarization of redistributed routes, tags from component routes SHOULD NOT be propagated to the summary. Implementations SHOULD provide a mechanism to configure tags for area ranges, NSSA ranges, and redistributed route summaries.

An OSPF router supporting this specification MUST be able to advertise and interpret one 32-bit tag for prefixes. An OSPF router supporting this specification MAY be able to advertise and propagate multiple 32-bit tags. The maximum tags that an implementation supports is a local matter depending upon supported applications using the prefix or link tags.

When a single tag is advertised for AS External or NSSA LSA prefix, the existing tag in OSPFv2 and OSPFv3 AS-External-LSA and NSSA-LSA encodings SHOULD be utilized. This will facilitate backward compatibility with implementations that do not support this specification.

##### 4.1. Equal-Cost Multipath Applicability

When multiple LSAs contribute to an OSPF route, it is possible that these LSAs will all have different tags. In this situation, the OSPF router MUST associate the tags from one of the LSAs contributing a path and, if the implementation supports multiple tags, MAY associate tags for multiple contributing LSAs up to the maximum number of tags supported.

#### 5. Security Considerations

This document describes a generic mechanism for advertising administrative tags for OSPF prefixes. The administrative tags are generally less critical than the topology information currently

advertised by the base OSPF protocol. The security considerations for the generic mechanism are dependent on the future application and, as such, should be described as additional capabilities are proposed for advertisement. Security considerations for the base OSPF protocol are covered in [RFC2328] and [RFC5340].

## 6. IANA Considerations

The following values should be allocated from the OSPF Extended Prefix TLV Sub-TLV Registry [RFC7684]:

- o TBD - 32-bit Administrative Tag TLV

The following values should be allocated from the OSPFv3 Extended-LSA Sub-TLV Registry [RFC8362]:

- o TBD - 32-bit Administrative Tag TLV

## 7. Acknowledgments

The authors of RFC 5130 are acknowledged since this document draws upon both the ISIS specification and deployment experience.

Thanks to Donnie Savage for his comments and questions.

The RFC text was produced using Marshall Rose's xml2rfc tool.

## 8. References

### 8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <<https://www.rfc-editor.org/info/rfc2328>>.
- [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>>.
- [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>>.

- [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>>.
- [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>>.

## 8.2. Informative References

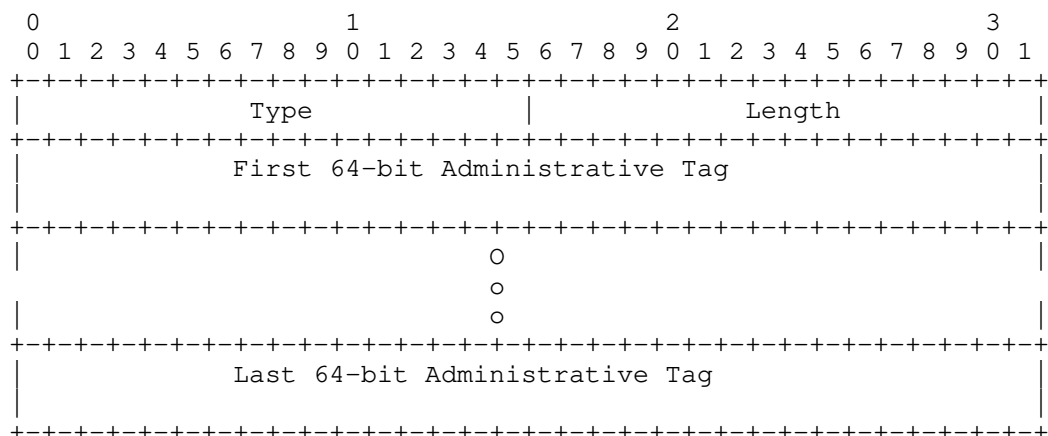
- [I-D.ietf-ospf-te-link-attr-reuse] Psenak, P., Ginsberg, L., Henderickx, W., Tantsura, J., and J. Drake, "OSPF Application-Specific Link Attributes", draft-ietf-ospf-te-link-attr-reuse-16 (work in progress), June 2020.
- [RFC3101] Murphy, P., "The OSPF Not-So-Stubby Area (NSSA) Option", RFC 3101, DOI 10.17487/RFC3101, January 2003, <<https://www.rfc-editor.org/info/rfc3101>>.
- [RFC5130] Previdi, S., Shand, M., Ed., and C. Martin, "A Policy Control Mechanism in IS-IS Using Administrative Tags", RFC 5130, DOI 10.17487/RFC5130, February 2008, <<https://www.rfc-editor.org/info/rfc5130>>.

## Appendix A. 64-Bit Administrative Tag Sub-TLV

The definition of the 64-bit tag was considered but discarded given that there is no strong requirement or use case. The specification is included here for information.

This sub-TLV will carry one or more 64-bit unsigned integer values that will be used as administrative tags.

The format of the 64-bit Administrative Tag TLV is as follows:



- Type** A 16-bit field set to TBD. The value MAY be different depending upon the registry from which it is allocated.
- Length** A 16-bit field that indicates the length of the value portion in octets and will be a multiple of 8 octets dependent on the number of administrative tags advertised. If the sub-TLV is specified, at least one administrative tag must be advertised.
- Value** A variable length list of one or more 64-bit administrative tags.

### 64-bit Administrative Tag TLV

## Appendix B. Link Administrative Tags

The advertisement of administrative tags corresponding to links has been removed from the document. The specification of advertising link administrative groups as specified in



[I-D.ietf-ospf-te-link-attr-reuse] advertising administrative tags  
for links.

Authors' Addresses

Acee Lindem (editor)  
Cisco Systems  
301 Midenhall Way  
Cary, NC 27513  
USA

EMail: [acee@cisco.com](mailto:acee@cisco.com)

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

EMail: [ppsenak@cisco.com](mailto:ppsenak@cisco.com)

Internet  
Internet-Draft  
Intended status: Informational  
Expires: February 14, 2020

A. Lindem  
S. Palani  
Cisco Systems  
Y. Qu  
Futurewei  
August 13, 2019

YANG Model for OSPFv3 Extended LSAs  
draft-acee-lsr-ospfv3-extended-lsa-yang-06

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 February 14, 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. Overview . . . . .	2
1.1. Requirements Language . . . . .	2
2. OSPFv3 Extended LSAs . . . . .	2
3. OSPFv3 Extended LSA Yang Module . . . . .	11
4. Security Considerations . . . . .	26
5. IANA Considerations . . . . .	27
6. Acknowledgements . . . . .	28
7. References . . . . .	28
7.1. Normative References . . . . .	28
7.2. Informative References . . . . .	29
Authors' Addresses . . . . .	29

## 1. Overview

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

This document defines a YANG data model 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].

### 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. 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 {extended-lsa-support}?
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area:
      +--rw extended-lsa-support?  boolean {extended-lsa-support}?
  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:
  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-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
+--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
+---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-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-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
    +--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
    |   +--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

```

### 3. OSPFv3 Extended LSA Yang Module

```

<CODE BEGINS> file "ietf-ospfv3-extended-lsa@2019-08-13.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";
  }

  import ietf-inet-types {
    prefix "inet";
    reference "RFC 6021 - 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 XXXX - 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:shpalani@cisco.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.

  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
  (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 2019-08-13 {
  description
    "Initial revision.";
```

```
    reference
      "RFC XXXX: A YANG Data Model for OSPFv3 Extended LSAs.";
  }

  feature extended-lsa-support {
    description
      "Support for OSPFv3 Extended LSAs";
    reference
      "RFC 8362 - OSPFv3 Link State Advertisement (LSA)
      Extensibility";
  }

  /*
  * OSPFv3 Extend LSA Type Identities
  */
  identity ospfv3-e-router-lsa {
    base ospf:ospfv3-lsa-type;
    description
      "OSPFv3 Extended Router LSA - Type 0xA021";
  }

  identity ospfv3-e-network-lsa {
    base ospf:ospfv3-lsa-type;
    description
      "OSPFv3 Extended Network LSA - Type 0xA022";
  }

  identity ospfv3-e-summary-lsa-type {
    base ospf:ospfv3-lsa-type;
    description
      "OSPFv3 Extended Summary LSA types";
  }

  identity ospfv3-e-inter-area-prefix-lsa {
    base ospfv3-e-summary-lsa-type;
    description
      "OSPFv3 Extended Inter-area Prefix LSA - Type 0xA023";
  }

  identity ospfv3-e-inter-area-router-lsa {
    base ospfv3-e-summary-lsa-type;
    description
      "OSPFv3 Extended Inter-area Router LSA - Type 0xA024";
  }

  identity ospfv3-e-external-lsa-type {
    base ospf:ospfv3-lsa-type;
    description
```

```
    "OSPFv3 Extended External LSA types";
}

identity ospfv3-e-as-external-lsa {
    base ospfv3-e-external-lsa-type;
    description
        "OSPFv3 Extended AS-External LSA - Type 0xC025";
}

identity ospfv3-e-nssa-lsa {
    base ospfv3-e-external-lsa-type;
    description
        "OSPFv3 Extended Not-So-Stubby-Area (NSSA) LSA -
        Type 0xA027";
}

identity ospfv3-e-link-lsa {
    base ospf:ospfv3-lsa-type;
    description
        "OSPFv3 Extended Link LSA - Type 0x8028";
}

identity ospfv3-e-intra-area-prefix-lsa {
    base ospf:ospfv3-lsa-type;
    description
        "OSPFv3 Extended Intra-area Prefix LSA - Type 0xA029";
}

identity ospfv3-e-prefix-option {
    description
        "Base identity for OSPFv3 Prefix Options.";
}

identity nu-bit {
    base ospfv3-e-prefix-option;
    description
        "When set, the prefix should be excluded
        from IPv6 unicast calculations.";
}

identity la-bit {
    base ospfv3-e-prefix-option;
    description
        "When set, the prefix is actually an IPv6 interface
        address of the Advertising Router.";
}

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.";
}

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.";
}

identity ospfv3-e-external-prefix-option {
    description
        "Base identity for OSPFv3 External Prefix Options.";
}

identity e-bit {
    base ospfv3-e-external-prefix-option;
    description
        "When set, the metric specified is a Type 2
        external metric.";
}

grouping unknown-sub-tlv {
    description
        "Unknown TLV grouping";
    container unknown-sub-tlv {
        uses ospf:tlv;
        description "Unknown External TLV Sub-TLV";
    }
}

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.";
}

leaf prefix-length {
    type uint8 {
        range "0..128";
    }
    description "Prefix length.";
}

}

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";
}

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";
}

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";
}

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";
}

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";
}

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";
}

grouping ipv4-link-local-tlv {
    container ipv4-link-local-tlv {
        description "IPv6 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";
}

grouping ospfv3-e-lsa-body {
    description "OSPFv3 Extended LSA body.";
    container e-router {
        when "derived-from ../../ospf:header/ospf:type, "
            + "'ospfv3-e-router-lsa'" {
            description "Only valid for OSPFv3 Extended-Router LSAs";
        }
        description "OSPFv3 Extended Router LSA";
        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 uint8;
            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 "derived-from(.../ospf:header/ospf:type, "
        + "'ospfv3-e-network-lsa') " {
        description
            "Only applies to E-Network LSAs.";
    }
    description "Extended Network LSA";
    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 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 "derived-from ../../ospf:header/ospf:type, "
    + "'ospfv3-e-inter-area-prefix-lsa'" {
    description
      "Only applies to E-Inter-Area-Prefix LSAs.";
  }
  description "Extended Inter-Area Prefix LSA";
  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 "derived-from ../../ospf:header/ospf:type, "
        + "'ospfv3-e-inter-area-router-lsa'" {
        description
            "Only applies to E-Inter-Area-Router LSAs.";
    }
    description "Extended Inter-Area Router LSA";
    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-as-external {  
    when "derived-from-or-self ../../ospf:header/ospf:type, "  
      + "'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."  
  }  
  
  container e-nssa {  
    when "derived-from-or-self ../../ospf:header/ospf:type, "  
      + "'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."  
  }  
  
  container e-link {  
    when "derived-from-or-self ../../ospf:header/ospf:type, "  
      + "'ospfv3-e-link-lsa'" {  
      description  
        "Only applies to Extended Link LSAs."  
    }  
    description "E-Link LSA";  
    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;
    }
  }

  container e-intra-area-prefix {
    when "derived-from-or-self ../../ospf:header/ospf:type, "
      + "'ospfv3-e-intra-area-prefix-lsa'" {
      description
        "Only applies to E-Intra-Area-Prefix LSAs.";
    }
    description "E-Intra-Area-Prefix LSA";
    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;
    }
  }
}

/* Configuration */
augment "/rt:routing/rt:control-plane-protocols"
  + "/rt:control-plane-protocol/ospf:ospf" {
```

```
    when "/rt:routing/rt:control-plane-protocols"
      + "/rt:control-plane-protocol/rt:type = 'ospf:ospfv3'" {
        description
          "This augments the OSPFv3 routing protocol when used.";
      }
    description
      "This augments the OSPFv3 protocol configuration
      with segment routing.";
    leaf extended-lsa-support {
      if-feature extended-lsa-support;
      type boolean;
      default false;
      description
        "Enable OSPFv3 Extended LSA Support for the OSPFv3
        domain";
    }
  }

augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area" {
  when "'ospf:.../.../.../.../rt:type' = 'ospf:ospfv3'" {
    description
      "This augments the OSPFv3 area configuration
      when used.";
  }
  description
    "This augments the OSPFv3 protocol area
    configuration with Extend LSA support";
  leaf extended-lsa-support {
    if-feature extended-lsa-support;
    type boolean;
    default false;
    description
      "Enable OSPFv3 Extended LSA Support for the OSPFv3 area";
  }
}

/*
 * 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:routing/rt:control-plane-protocols"
```

```

    + "/rt:control-plane-protocol/rt:type = 'ospf:ospfv3'" {
      description
        "This augmentation is only valid for OSPFv3.";
    }
    description
      "OSPFv3 Link-Scoped Extended LSAs";
  }

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
        E-Router LSAs";
    }
  uses ospfv3-e-lsa-body;
  description
    "OSPFv3 Area-Scoped Extended LSAs";
}

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 "'ospf:../.../.../.../.../.../.../.../.../...'"
    + "rt:type' = 'ospf:ospfv3'" {
      description
        "This augmentation is only valid for OSPFv3.";
    }
  uses ospfv3-e-lsa-body;
  description
    "OSPFv3 AS-Scoped Extended LSAs";
}
}

<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 [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 `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. 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. 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.

## 5. 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
```

## 6. 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.

## 7. References

### 7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [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>>.
- [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>>.

- [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>>.
- [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>>.

## 7.2. Informative References

- [I-D.ietf-ospf-yang]  
Yeung, D., Qu, Y., Zhang, Z., Chen, I., and A. Lindem,  
"YANG Data Model for OSPF Protocol", draft-ietf-ospf-  
yang-26 (work in progress), August 2019.

## Authors' Addresses

Acee Lindem  
Cisco Systems  
301 Midenhall Way  
Cary, NC 27513

EMail: [acee@cisco.com](mailto:acee@cisco.com)

Sharmila Palani  
Cisco Systems  
170 West Tasman Drive  
San Jose, CA 95134

EMail: [shpalani@cisco.com](mailto:shpalani@cisco.com)

Yingzhen Qu  
Futurewei  
2330 Central Expressway  
Santa Clara, CA 95050  
USA

EMail: yingzhen.qu@futurewei.com

Networking Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: September 7, 2019

P. Psenak, Ed.  
C. Filsfils  
Cisco Systems  
A. Bashandy  
Individual  
B. Decraene  
Orange  
Z. Hu  
Huawei Technologies  
March 6, 2019

IS-IS Extensions to Support Routing over IPv6 Dataplane  
draft-bashandy-isis-srv6-extensions-05.txt

Abstract

Segment Routing (SR) allows for a flexible definition of end-to-end paths by encoding paths as sequences of topological sub-paths, called "segments". Segment routing architecture can be implemented over an MPLS data plane as well as an IPv6 data plane. This draft describes the IS-IS extensions required to support Segment Routing over an IPv6 data plane.

Requirements Language

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

Status of This Memo

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 September 7, 2019.



## 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 . . . . .	3
2. SRv6 Capabilities sub-TLV . . . . .	3
3. Advertising Supported Algorithms . . . . .	4
4. Advertising Maximum SRv6 SID Depths . . . . .	4
4.1. Maximum Segments Left MSD Type . . . . .	5
4.2. Maximum End Pop MSD Type . . . . .	5
4.3. Maximum T.Insert MSD Type . . . . .	5
4.4. Maximum T.Encaps MSD Type . . . . .	5
4.5. Maximum End D MSD Type . . . . .	6
5. SRv6 SIDs and Reachability . . . . .	6
6. Advertising Locators and End SIDs . . . . .	7
6.1. SRv6 Locator TLV Format . . . . .	8
6.2. SRv6 End SID sub-TLV . . . . .	9
7. Advertising SRv6 End.X SIDs . . . . .	11
7.1. SRv6 End.X SID sub-TLV . . . . .	11
7.2. SRv6 LAN End.X SID sub-TLV . . . . .	13
8. Advertising Endpoint Behaviors . . . . .	14
9. IANA Considerations . . . . .	15
9.1. SRv6 Locator TLV . . . . .	15
9.1.1. SRv6 End SID sub-TLV . . . . .	15
9.1.2. Revised sub-TLV table . . . . .	16
9.2. SRv6 Capabilities sub-TLV . . . . .	16
9.3. SRv6 End.X SID and SRv6 LAN End.X SID sub-TLVs . . . . .	17
9.4. MSD Types . . . . .	17
10. Security Considerations . . . . .	17
11. Contributors . . . . .	17
12. References . . . . .	18
12.1. Normative References . . . . .	18
12.2. Informative References . . . . .	20
Authors' Addresses . . . . .	21

## 1. Introduction

With Segment Routing (SR) [I-D.ietf-spring-segment-routing], 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 [I-D.ietf-6man-segment-routing-header]. SRv6 refers to this SR instantiation on the IPv6 dataplane.

The network programming paradigm [I-D.filsfils-spring-srv6-network-programming] is central to SRv6. It describes how any function can be bound to a SID and how any network program can be expressed as a combination of SID's.

This document specifies IS-IS extensions that allow the IS-IS protocol to encode some of these functions.

Familiarity with the network programming paradigm [I-D.filsfils-spring-srv6-network-programming] is necessary to understand the extensions specified in this document.

This document defines one new top level IS-IS TLV and several new IS-IS sub-TLVs.

The SRv6 Capabilities sub-TLV announces the ability to support SRv6 and some Endpoint functions listed in Section 7 as well as advertising limitations when applying such Endpoint functions.

The SRv6 Locator top level TLV announces SRv6 locators - a form of summary address for the set of topology/algorithm specific SIDs associated with a node.

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 function is bound to each instantiated SID.

## 2. SRv6 Capabilities sub-TLV

A node indicates that it has support for SRv6 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: Suggested value 25, to be assigned by IANA

Length: 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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| |O| | | | | | | | | | | | | | | | | | | | | | | | | | | |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

where:

O-flag: If set, the router supports use of the O-bit in the Segment Routing Header(SRH) as defined in [I-D.ali-spring-srv6-oam].

### 3. Advertising Supported Algorithms

SRv6 capable router indicates supported algorithm(s) by advertising the SR Algorithm TLV as defined in [I-D.ietf-isis-segment-routing-extensions].

### 4. Advertising Maximum SRv6 SID Depths

[I-D.ietf-isis-segment-routing-msd] 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, 141, 222, and 223.

This document defines the relevant SRv6 MSDs and requests MSD type assignments in the MSD Types registry created by [I-D.ietf-isis-segment-routing-msd].

#### 4.1. Maximum Segments Left MSD Type

The Maximum Segments Left MSD Type specifies the maximum value of the "SL" field [I-D.ietf-6man-segment-routing-header] in the SRH of a received packet before applying the Endpoint function associated with a SID.

SRH Max SL Type: 41 (Suggested value - to be assigned by IANA)

If no value is advertised the supported value is assumed to be 0.

#### 4.2. Maximum End Pop MSD Type

The Maximum End Pop MSD Type specifies the maximum number of SIDs in the top SRH in an SRH stack to which the router can apply "PSP" or "USP" as defined in [I-D.filsfils-spring-srv6-network-programming] flavors.

SRH Max End Pop Type: 42 (Suggested value - to be assigned by IANA)

If the advertised value is zero or no value is advertised then it is assumed that the router cannot apply PSP or USP flavors.

#### 4.3. Maximum T.Insert MSD Type

The Maximum T.Insert MSD Type specifies the maximum number of SIDs that can be inserted as part of the "T.insert" behavior as defined in [I-D.filsfils-spring-srv6-network-programming].

SRH Max T.insert Type: 43 (Suggested value - to be assigned by IANA)

If the advertised value is zero or no value is advertised then the router is assumed not to support any variation of the "T.insert" behavior.

#### 4.4. Maximum T.Encaps MSD Type

The Maximum T.Encaps MSD Type specifies the maximum number of SIDs that can be included as part of the "T.Encaps" behavior as defined in [I-D.filsfils-spring-srv6-network-programming] .

SRH Max T.encaps Type: 44 (Suggested value - to be assigned by IANA)

If the advertised value is zero then the router can apply T.Encaps only by encapsulating the incoming packet in another IPv6 header without SRH the same way IPinIP encapsulation is performed.

If the advertised value is non-zero then the router supports both IPinIP and SRH encapsulation subject to the SID limitation specified by the advertised value.

#### 4.5. Maximum End D MSD Type

The Maximum End D MSD Type specifies the maximum number of SIDs in an SRH when performing decapsulation associated with "End.Dx" functions (e.g., "End.DX6" and "End.DT6") as defined in [I-D.filsfils-spring-srv6-network-programming].

SRH Max End D Type: 45 (Suggested value - to be assigned by IANA)

If the advertised value is zero or no value is advertised then it is assumed that the router cannot apply "End.DX6" or "End.DT6" functions if the extension header right underneath the outer IPv6 header is an SRH.

#### 5. SRv6 SIDs and Reachability

As discussed in [I-D.filsfils-spring-srv6-network-programming], an SRv6 Segment Identifier (SID) is 128 bits and represented as

LOC:FUNCT

where LOC (the locator portion) is the L most significant bits and FUNCT is the 128-L least significant bits. L is called the locator length and is flexible. Each operator is free to use the locator length it chooses.

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 6.1). 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.

Locators are routable and MAY also be advertised in Prefix Reachability TLVs (236 or 237).

Locators associated with algorithm 0 (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 SRv6 traffic.

In cases where a locator advertisement is received in both in 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 on SRv6 capable/SRv6 incapable routers.

SRv6 SIDs are advertised as sub-TLVs in the SRv6 Locator TLV except for SRv6 End.X SIDs/LAN End.X SIDs which are associated with a specific Neighbor/Link and are therefore advertised as sub-TLVs in TLVs 22, 23, 222, 223, and 141.

SRv6 SIDs 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. There are a number of ways in which this requirement could be compromised

- o Another locator associated with a different topology/algorithm is the longest match
- o A prefix advertisement (i.e., from TLV 236 or 237) is the longest match

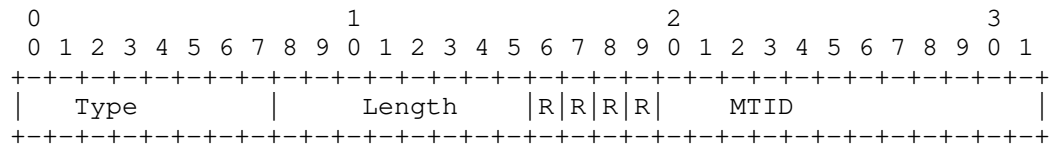
## 6. Advertising Locators and End SIDs

The SRv6 Locator TLV is introduced to advertise SRv6 Locators and End SIDs associated with each locator.

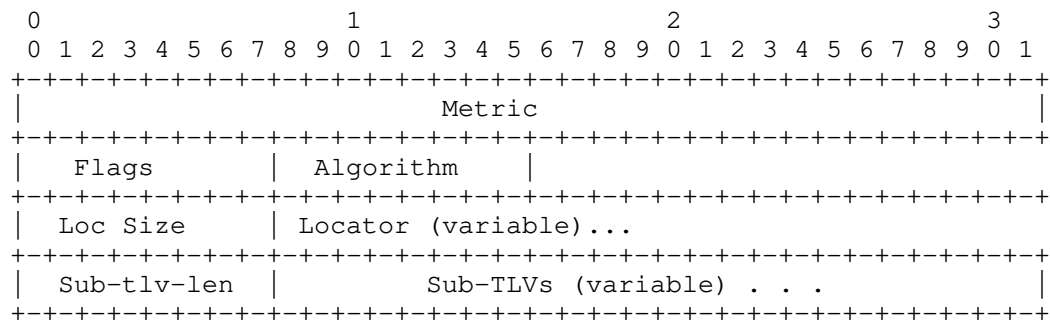
This new TLV shares the sub-TLV space defined for TLVs 135, 235, 236 and 237.

## 6.1. SRv6 Locator TLV Format

The SRv6 Locator TLV has the following format:



Followed by one or more locator entries of the form:



Type: 27 (Suggested value to be assigned by IANA)

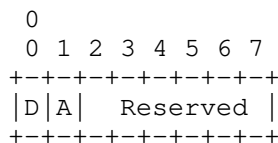
Length: variable.

MTID: Multitopology Identifier as defined in [RFC5120].  
Note that the value 0 is legal.

Locator entry:

Metric: 4 octets. As described in [RFC5305].

Flags: 1 octet. The following flags are defined



where:

D bit: When the Locator is leaked from level-2 to level-1, the D bit MUST be set. Otherwise, this bit MUST be clear. Locators with the D bit set MUST NOT be leaked from level-1 to level-2.

This is to prevent looping.

A bit: When the Locator is configured as anycast, the A bit SHOULD be set. Otherwise, this bit MUST be clear.

The remaining bits are reserved for future use. They SHOULD be set to zero on transmission and MUST be ignored on receipt.

Algorithm: 1 octet. Associated algorithm. Algorithm values are defined in the IGP Algorithm Type registry.

Loc-Size: 1 octet. Number of bits in the Locator field.  
(1 - 128)

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.

Sub-TLV-length: 1 octet. Number of octets used by sub-TLVs

Optional sub-TLVs.

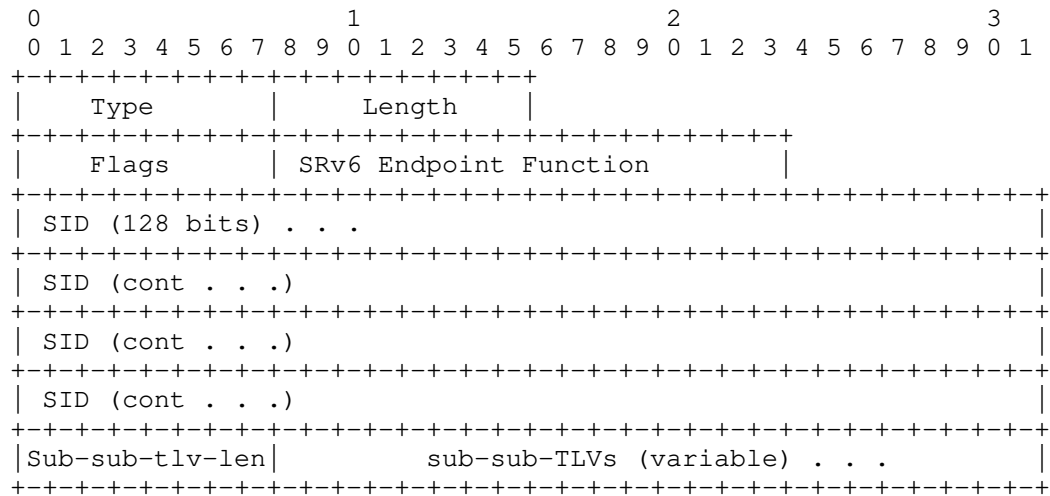
## 6.2. SRv6 End SID sub-TLV

The SRv6 End SID sub-TLV is introduced to advertise SRv6 Segment Identifiers (SID) with Endpoint functions which do not require a particular neighbor in order to be correctly applied [I-D.filsfils-spring-srv6-network-programming]. SRv6 SIDs associated with a neighbor are advertised using the sub-TLVs defined in Section 6.

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 (Suggested value to be assigned by IANA)

Length: variable.

Flags: 1 octet. No flags are currently defined.

SRv6 Endpoint Function: 2 octets. As defined in  
[I-D.filsfils-spring-srv6-network-programming]  
Legal function values for this sub-TLV are defined in Section 7.

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

The SRv6 End SID MUST be a subnet of the associated Locator. SRv6 End SIDs which are NOT a subnet of 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.

## 7. Advertising SRv6 End.X SIDs

Certain SRv6 Endpoint functions

[I-D.filsfils-spring-srv6-network-programming] must be associated with a particular neighbor, and in case of multiple layer 3 links to the same neighbor, with a particular link in order to be correctly applied.

This document defines two new sub-TLVs of TLV 22, 23, 222, 223, and 141 - namely "SRv6 End.X SID" and "SRv6 LAN End.X SID".

IS-IS Neighbor advertisements are topology specific - but not algorithm specific. End.X SIDs therefore inherit the topology from the associated neighbor advertisement, but the algorithm is specified in the individual SID.

All End.X SIDs MUST be a subnet of a Locator with matching topology and algorithm which is advertised by the same node in an SRv6 Locator TLV. End.X SIDs which do not meet this requirement MUST be ignored.

### 7.1. 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:

```

      0                               1                               2                               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|  Type      |      Length      |                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|  Flags     |      Algorithm   |      Weight      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|  SRv6 Endpoint Function  |
+-----+-----+-----+-----+-----+-----+-----+-----+
|  SID (128 bits) . . .   |
+-----+-----+-----+-----+-----+-----+-----+-----+
|  SID (cont . . .)      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|  SID (cont . . .)      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|  SID (cont . . .)      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|Sub-sub-tlv-len|      Sub-sub-TLVs (variable) . . .   |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Type: 43 (Suggested value to be assigned by IANA)

Length: variable.

Flags: 1 octet.

```

    0 1 2 3 4 5 6 7
    +-+-+-+-+-+-+-+-+
    |B|S|P|Reserved |
    +-+-+-+-+-+-+-+-+

```

where:

B-Flag: Backup flag. If set, the End.X SID is eligible for protection (e.g., using IPFRR) as described in [RFC8355].

S-Flag. Set flag. When set, the S-Flag indicates that the End.X 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 End.X SID is persistently allocated, i.e., the End.X SID value remains consistent across router restart and/or interface flap.

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

Algorithm: 1 octet. Associated algorithm. Algorithm values are defined in the IGP Algorithm Type registry.

Weight: 1 octet. The value represents the weight of the End.X SID for the purpose of load balancing. The use of the weight is defined in [I-D.ietf-spring-segment-routing].

SRv6 Endpoint Function: 2 octets. As defined in [I-D.filsfils-spring-srv6-network-programming]  
Legal function values for this sub-TLV are defined in Section 7.

SID: 16 octets. This field encodes the advertised SRv6 SID.

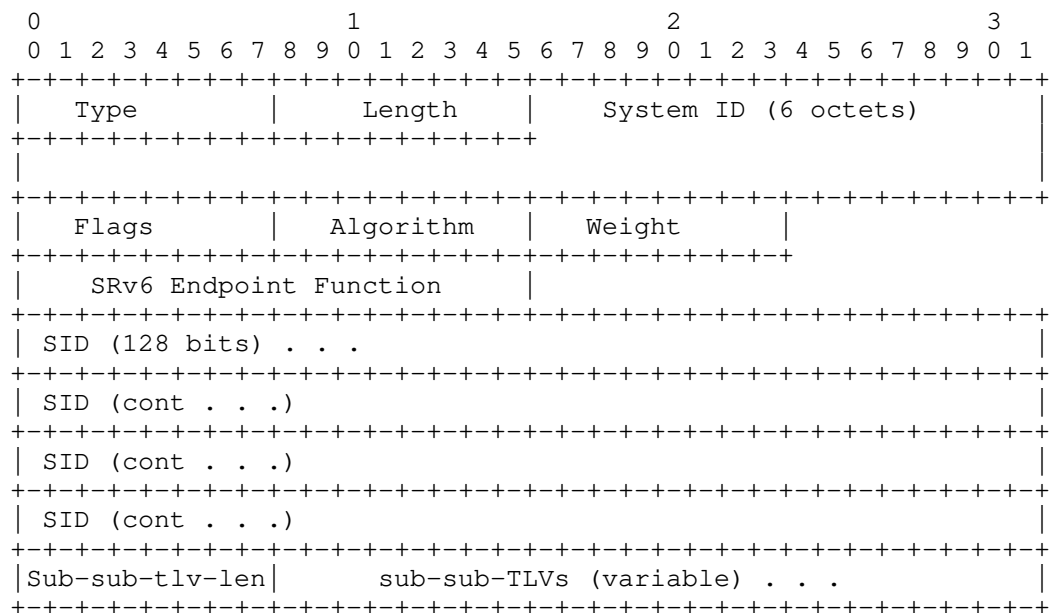
Sub-sub-TLV-length: 1 octet. Number of octets used by sub-sub-TLVs

Note that multiple TLVs for the same neighbor may be required in order to advertise all of the SRv6 End.X SIDs associated with that neighbor.

## 7.2. 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 a large number of neighbors may exist on a given LAN a large number of 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 of the SRv6 End.X SIDs associated with that neighbor.

The SRv6 LAN End.X SID sub-TLV has the following format:

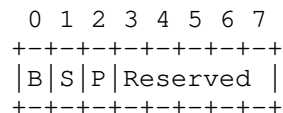


Type: 44 (Suggested value to be assigned by IANA)

Length: variable.

System-ID: 6 octets of 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 6.1.  
Other bits: MUST be zero when originated and ignored when received.

Algorithm: 1 octet. Associated algorithm. Algorithm values are defined in the IGP Algorithm Type registry.

Weight: 1 octet. The value represents the weight of the End.X SID for the purpose of load balancing. The use of the weight is defined in [I-D.ietf-spring-segment-routing].

SRv6 Endpoint Function: 2 octets. As defined in [I-D.filsfils-spring-srv6-network-programming]  
Legal function values for this sub-TLV are defined in Section 7.

SID: 16 octets. This field encodes the advertised SRv6 SID.

Sub-sub-TLV-length: 1 octet. Number of octets used by sub-sub-TLVs.

## 8. Advertising Endpoint Behaviors

Endpoint behaviors are defined in [I-D.filsfils-spring-srv6-network-programming] and [I-D.ali-spring-srv6-oam]. The numerical identifiers for the Endpoint behaviors are defined in the "SRv6 Endpoint Behaviors" registry defined in [I-D.filsfils-spring-srv6-network-programming]. This section lists the Endpoint behaviors and their identifiers, which MAY be advertised by IS-IS and the SID sub-TLVs in which each type MAY appear.

Endpoint Behavior	Endpoint Behavior Identifier	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.T (PSP, USP, USD)	9-12, 36-39	Y	N	N
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.DT64	20	Y	N	N
End.OP	40	Y	N	N
End.OTP	41	Y	N	N

## 9. IANA Considerations

This document requests allocation for the following TLVs, sub-TLVs, and sub-sub-TLVs as well updating the ISIS TLV registry and defining a new registry.

### 9.1. SRv6 Locator TLV

This document adds one new TLV to the IS-IS TLV Codepoints registry.

Value: 27 (suggested - to be assigned by IANA)

Name: SRv6 Locator

This TLV shares sub-TLV space with existing "Sub-TLVs for TLVs 135, 235, 236 and 237 registry". The name of this registry needs to be changed to "Sub-TLVs for TLVs 27, 135, 235, 236 and 237 registry".

#### 9.1.1. SRv6 End SID sub-TLV

This document adds the following new sub-TLV to the (renamed) "Sub-TLVs for TLVs 27, 135, 235, 236 and 237 registry".

Value: 5 (suggested - to be assigned by IANA)

Name: SRv6 End SID

This document requests the creation of a new IANA managed registry for sub-sub-TLVs of the SRv6 End SID sub-TLV. The registration procedure is "Expert Review" as defined in [RFC7370]. Suggested registry name is "sub-sub-TLVs for SRv6 End SID sub-TLV". No sub-sub-TLVs are defined by this document except for the reserved value.

0: Reserved

1-255: Unassigned

#### 9.1.2. Revised sub-TLV table

The revised table of sub-TLVs for the (renamed) "Sub-TLVs for TLVs 27, 135, 235, 236 and 237 registry" is shown below:

Type	27	135	235	236	237
1	n	y	y	y	y
2	n	y	y	y	y
3	n	y	y	y	y
4	y	y	y	y	y
5	y	n	n	n	n
11	y	y	y	y	y
12	y	y	y	y	y

#### 9.2. SRv6 Capabilities sub-TLV

This document adds the definition of a new sub-TLV in the "Sub- TLVs for TLV 242 registry".

Type: 25 (Suggested - to be assigned by IANA)

Description: SRv6 Capabilities

This document requests the creation of a new IANA managed registry for sub-sub-TLVs of the SRv6 Capability sub-TLV. The registration procedure is "Expert Review" as defined in [RFC7370]. Suggested registry name is "sub-sub-TLVs for SRv6 Capability sub-TLV". No sub-sub-TLVs are defined by this document except for the reserved value.

0: Reserved

1-255: Unassigned

### 9.3. SRv6 End.X SID and SRv6 LAN End.X SID sub-TLVs

This document adds the definition of two new sub-TLVs in the "sub-TLVs for TLV 22, 23, 25, 141, 222 and 223 registry".

Type: 43 (suggested - to be assigned by IANA)

Description: SRv6 End.X SID

Type: 44 (suggested - to be assigned by IANA)

Description: SRv6 LAN End.X SID

Type	22	23	25	141	222	223
------	----	----	----	-----	-----	-----

43	Y	Y	Y	Y	Y	Y
44	Y	Y	Y	Y	Y	Y

### 9.4. MSD Types

This document defines the following new MSD types. These types are to be defined in the IGP MSD Types registry defined in [I-D.ietf-isis-segment-routing-msd] .

All values are suggested values to be assigned by IANA.

Type	Description
------	-------------

41	SRH Max SL
42	SRH Max End Pop
43	SRH Max T.insert
44	SRH Max T.encaps
45	SRH Max End D

## 10. Security Considerations

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

## 11. 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

## 12. References

### 12.1. Normative References

[I-D.ali-spring-srv6-oam]

Ali, Z., Filsfils, C., Kumar, N., Pignataro, C.,  
faiqbal@cisco.com, f., Gandhi, R., Leddy, J., Matsushima,  
S., Raszuk, R., daniel.voyer@bell.ca, d., Dawra, G.,  
Peirens, B., Chen, M., and G. Naik, "Operations,  
Administration, and Maintenance (OAM) in Segment Routing  
Networks with IPv6 Data plane (SRv6)", draft-ali-spring-  
srv6-oam-02 (work in progress), October 2018.

[I-D.filsfils-spring-srv6-network-programming]

Filsfils, C., Camarillo, P., Leddy, J.,  
daniel.voyer@bell.ca, d., Matsushima, S., and Z. Li, "SRv6  
Network Programming", draft-filsfils-spring-srv6-network-  
programming-07 (work in progress), February 2019.

[I-D.ietf-6man-segment-routing-header]

Filsfils, C., Previdi, S., Leddy, J., Matsushima, S., and  
d. daniel.voyer@bell.ca, "IPv6 Segment Routing Header  
(SRH)", draft-ietf-6man-segment-routing-header-16 (work in  
progress), February 2019.

[I-D.ietf-isis-segment-routing-extensions]

Previdi, S., Ginsberg, L., Filsfils, C., Bashandy, A.,  
Gredler, H., and B. Decraene, "IS-IS Extensions for  
Segment Routing", draft-ietf-isis-segment-routing-  
extensions-22 (work in progress), December 2018.

[I-D.ietf-isis-segment-routing-msd]

Tantsura, J., Chunduri, U., Aldrin, S., and L. Ginsberg,  
"Signaling MSD (Maximum SID Depth) using IS-IS", draft-  
ietf-isis-segment-routing-msd-19 (work in progress),  
October 2018.

[ISO10589]

Standardization", I. "O. F., "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,  
<<http://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>>.
- [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>>.
- [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>>.
- [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>>.

## 12.2. Informative References

- [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.
- [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

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: December 7, 2020

H. Chen  
Futurewei  
M. Toy  
Verizon  
Y. Yang  
IBM  
A. Wang  
China Telecom  
X. Liu  
Volta Networks  
Y. Fan  
Casa Systems  
L. Liu  
Fujitsu  
June 5, 2020

Flooding Topology Computation Algorithm  
draft-cc-lsr-flooding-reduction-09

Abstract

This document proposes an algorithm for a node to compute a flooding topology, which is a subgraph of the complete topology per underline physical network. When every node in an area automatically calculates a flooding topology by using a same algorithm and floods the link states using the flooding topology, the amount of flooding traffic in the network is greatly reduced. This would reduce convergence time with a more stable and optimized routing environment.

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 December 7, 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. Introduction . . . . .	2
2. Terminology . . . . .	3
3. Flooding Topology . . . . .	3
3.1. Flooding Topology Construction . . . . .	4
4. Algorithms to Compute Flooding Topology . . . . .	4
4.1. Algorithm with Considering Degree . . . . .	5
4.2. Algorithm with Considering Others . . . . .	6
5. Security Considerations . . . . .	6
6. IANA Considerations . . . . .	6
7. Acknowledgements . . . . .	7
8. References . . . . .	7
8.1. Normative References . . . . .	7
8.2. Informative References . . . . .	7
Appendix A. FT Computation Details through Example . . . . .	7
Authors' Addresses . . . . .	11

## 1. Introduction

For some networks such as dense Data Center (DC) networks, the existing Link State (LS) flooding mechanism is not efficient and may have some issues. The extra LS flooding consumes network bandwidth. Processing the extra LS flooding, including receiving, buffering and decoding the extra LSs, wastes memory space and processor time. This

may cause scalability issues and affect the network convergence negatively.

This document proposes an algorithm for a node to compute a flooding topology, which is a subgraph of the complete topology per underline physical network. The physical network can be any network, including clos leaf spine network. It can be used in the distributed mode of flooding topology computation for flooding reduction and the centralized mode, which are described in [I-D.ietf-lsr-dynamic-flooding]. When the distributed mode is selected, every node in an area automatically calculates a flooding topology by using a same algorithm and floods the link states using the flooding topology, the amount of flooding traffic in the network is greatly reduced. This would reduce convergence time with a more stable and optimized routing environment.

There may be multiple algorithms for computing a flooding topology. Users can select one they prefer, and smoothly switch from one to another.

## 2. Terminology

LSA: A Link State Advertisement in OSPF.

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

LS: A Link Sate, which is an LSA or LSP.

FT: Flooding Topology.

FTC: Flooding Topology Computation.

## 3. Flooding Topology

For a given network topology, a flooding topology is a sub-graph or sub-network of the given network topology that has the same reachability to every node as the given network topology. Thus all the nodes in the given network topology MUST be in the flooding topology. All the nodes MUST be inter-connected directly or indirectly. As a result, LS flooding will in most cases occur only on the flooding topology, that includes all nodes but a subset of links. Note even though the flooding topology is a sub-graph of the original topology, any single LS MUST still be disseminated in the entire network.

### 3.1. Flooding Topology Construction

Many different flooding topologies can be constructed for a given network topology. For example, a chain connecting all the nodes in the given network topology is a flooding topology. A circle connecting all the nodes is another flooding topology. A tree connecting all the nodes is a flooding topology. In addition, the tree plus the connections between some leaves of the tree and branch nodes of the tree is a flooding topology.

The following parameters need to be considered for constructing a flooding topology:

- o Degree: The degree of the flooding topology is the maximum degree among the degrees of the nodes on the flooding topology. The degree of a node on the flooding topology is the number of connections on the flooding topology it has to other nodes.
- o Number of links: The number of links on the flooding topology is a key factor for reducing the amount of LS flooding. In general, the smaller the number of links, the less the amount of LS flooding.
- o Diameter: The diameter of the flooding topology is the shortest distance between the two most distant nodes on the flooding topology. It is a key factor for reducing the network convergence time. The smaller the diameter, the less the convergence time.
- o Redundancy: The redundancy of the flooding topology means a tolerance to the failures of some links and nodes on the flooding topology. If the flooding topology is split by some failures, it is not tolerant to these failures. In general, the larger the number of links on the flooding topology is, the more tolerant the flooding topology to failures.

Note that the flooding topology constructed by a node is dynamic in nature, that means when the base topology (the entire topology graph) changes, the flooding topology (the sub-graph) MUST be re-computed/re-constructed to ensure that any node that is reachable on the base topology MUST also be reachable on the flooding topology.

### 4. Algorithms to Compute Flooding Topology

There are many algorithms to compute a flooding topology. A simple and efficient one is briefed, which comprises:

- o Selecting a node R0 with the smallest node ID;



- o Building a tree using R0 as root in breadth first; and then
- o Connecting each node whose degree is one to another node to have a flooding topology.

#### 4.1. Algorithm with Considering Degree

The algorithm is described below, where a variable MaxD with an initial value 3, data structures candidate queue Cq and flooding topology FT are used. Cq and FT comprise elements of form (N, D, PHs), where N represents a Node, D is the Degree of node N, and PHs contains the Previous Hops of node N. The detailed FT computation by the algorithm is illustrated in Appendix A through an example.

The algorithm starts from node R0 as root with a maximum degree MaxD of value 3, a candidate queue  $Cq = \{(R0, D = 0, PHs = \{ \})\}$ , and an empty flooding topology  $FT = \{ \}$ . Cq contains one element (R0, D = 0, PHs = { }), where node R0 is the root, D = 0 indicates that the Degree (D for short) of R0 is 0 (i.e., the number of links on the flooding topology connected to R0 is 0), PHs = { } indicates that the Previous Hops (PHs for short) of R0 is empty.

1. Finding and removing the first element with node A in Cq that is not on FT and one PH's D in PHs < MaxD.

If A is root R0, then add the element into FT

otherwise (i.e.,  $A \neq R0$  with one PH's D in PHs < MaxD. Assume that PH is the first one in PHs whose D < MaxD), PH's D++, and add A with D = 1 and PHs = {PH} into FT.

Note: if no element in Cq satisfies the conditions, algorithm is restarted from R0, ++MaxD,  $Cq = \{(R0, D=0, PHs=\{ \})\}$ ,  $FT = \{ \}$ ;

2. If all the nodes are on the FT, then goto step 4;
3. Suppose that node Xi (i = 1, 2, ..., n) is connected to node A and not on FT, and X1, X2, ..., Xn are in an increasing order by their IDs (i.e., X1's ID < X2's ID < ... < Xn's ID). If Xi is not in Cq, then add it into the end of Cq with D = 0 and PHs = {A}; otherwise (i.e., Xi is in Cq), add A into the end of Xi's PHs; Goto step 1.
4. For each node B on FT whose D is one (from minimum to maximum node ID), find a link L attached to B such that L's remote node R has minimum D and ID, add link L between B and R into FT and increase B's D and R's D by one. Return FT.

#### 4.2. Algorithm with Considering Others

There may be some constraints on some nodes in a network. For example, in a spine-and-leaf network, there may be a constraint on the degree of every leaf node on the flooding topology, which is that the degree of every leaf node is not greater than a given number ConMaxD of value 2. For each of the other nodes such as the spine nodes, there is no such constraint, that is that ConMaxD is a huge number for each of these nodes.

Step 1 of the algorithm described above is updated below to consider this constraint. In addition to checking constraint PH's  $D < \text{MaxD}$ , step 1 checks another constraint PH's  $D < \text{PH's ConMaxD}$ .

1. Finding and removing the first element with node A in Cq that is not on FT and one PH's D in PHs  $< \text{MaxD}$  and PH's  $D < \text{PH's ConMaxD}$ .

If A is root R0, then add the element into FT

otherwise (i.e.,  $A \neq R0$  with one PH's D in PHs  $< \text{MaxD}$  and PH's  $D < \text{PH's ConMaxD}$ . Assume that PH is the first one in PHs whose  $D < \text{MaxD}$  and PH's  $D < \text{PH's ConMaxD}$ ), PH's  $D++$ , and add A with  $D = 1$  and PHs = {PH} into FT.

Note: if no element in Cq satisfies the conditions, algorithm is restarted from R0,  $++\text{MaxD}$ ,  $\text{Cq} = \{(R0, D=0, \text{PHs}=\{\})\}$ ,  $\text{FT} = \{\}$ ;

#### 5. Security Considerations

This document does not introduce any new security issue.

#### 6. IANA Considerations

Under Registry Name: "IGP Algorithm Type For Computing Flooding Topology" under an existing "Interior Gateway Protocol (IGP Parameters" IANA registries (refer to Section 7.3. IGP [I-D.ietf-lsr-dynamic-flooding]), IANA is requested to assign one value of IGP Algorithm Type For Computing Flooding Topology as follows:

Type Value	Type Name	reference
1	Breadth First Minimum Degree Algorithm	This document
2	Breadth First Leaf Constraint Algorithm	This document

## 7. Acknowledgements

The authors would like to thank Dean Cheng, Acee Lindem, Zhibo Hu, Robin Li, Stephane Litkowski and Alvaro Retana for their valuable suggestions and comments on this draft.

## 8. References

### 8.1. 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-06 (work in progress), May 2020.
- [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>>.
- [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <<https://www.rfc-editor.org/info/rfc2328>>.

### 8.2. Informative References

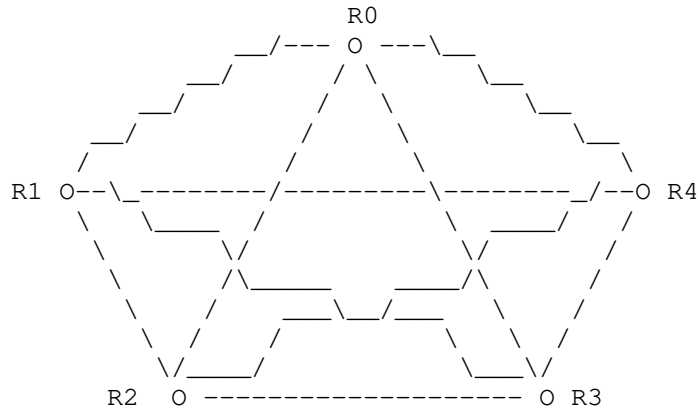
- [I-D.ietf-rtgwg-spf-uloop-pb-statement]  
Litkowski, S., Decraene, B., and M. Horneffer, "Link State protocols SPF trigger and delay algorithm impact on IGP micro-loops", draft-ietf-rtgwg-spf-uloop-pb-statement-10 (work in progress), January 2019.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines 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>>.

## Appendix A. FT Computation Details through Example

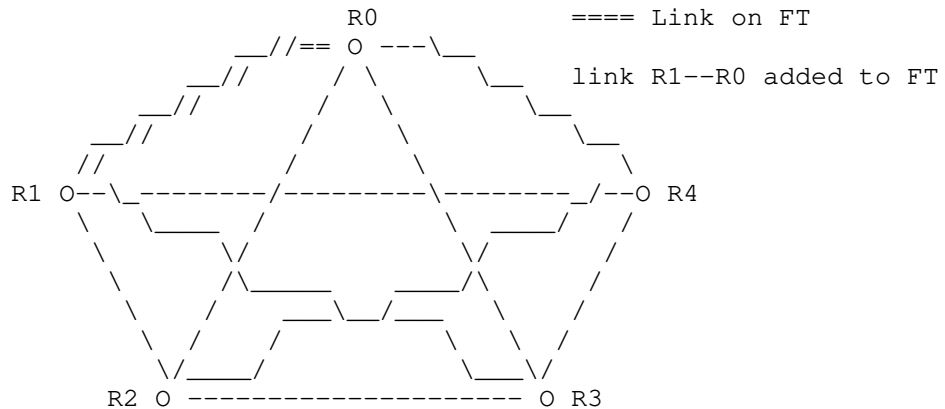
This section presents the details on FT computation by the algorithm through an example. The detailed procedure of computing a FT for a network of five nodes with full mesh connections is illustrated. Suppose that the network has five nodes R0, R1, R2, R3 and R4; R0's

ID < R1's ID < R2's ID < R3's ID < R4's ID. The algorithm starts with Cq = {(R0, D=0, PHs={})}, FT = {}, MaxD = 3.

```
0. // remove the first element containing root R0 from Cq
   Cq = { };
   // add the element into FT
   FT = { (R0,D=0,PHs={ }) }; // root R0 on FT
   // for each Ri connected to R0 (not in Cq), add it to the end of Cq
   Cq = { (R1,D=0,PHs={R0}), (R2,D=0,PHs={R0}), (R3,D=0,PHs={R0}),
          ^^^^^^^^^^^^^^^^^ (R4,D=0,PHs={R0}) }
```



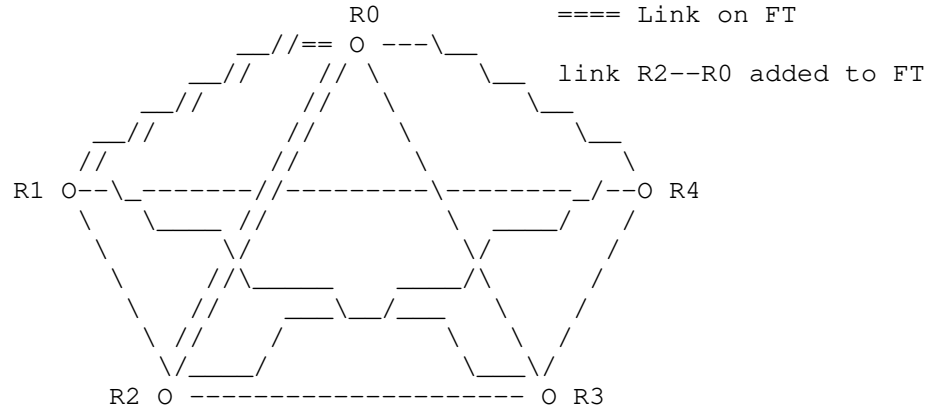
```
1. //remove first element (R1,D=0,PHs={R0}) from Cq, R0's D=0 < MaxD
   Cq = { (R2,0,{R0}), (R3,0,{R0}), (R4,0,{R0}) };
   // add (R1,1,{R0}) into FT, increase PH R0's D by one
   FT = { (R0,1, { }), (R1,1, {R0}) }; // Link R1--R0 on FT
          ^^^          ^^^^^^^^^^^^^
   // for Ri connected to R1 (in Cq) not on FT, append R1 to Ri's PHs
   Cq = { (R2,0, {R0,R1}), (R3,0, {R0,R1}), (R4,0,{R0,R1}) }.
          ^^          ^^          ^^
```



```

2. // remove the first element (R2,0, {R0,R1}) from Cq, R0's D=1 < MaxD
   Cq = { (R3,0, {R0,R1}), (R4,0,{R0,R1}) }
   // add (R2,1,{R0}) into FT, increase R0's D by one
   FT = { (R0,2,{  }), (R1,1,{R0}), (R2,1,{R0}) } //Link R2--R0 on FT
           ^^^               ^^^^^^^^^^^
   // for Ri connected to R2 (in Cq) not on FT, append R2 to Ri's PHs
   Cq = { (R3,0, {R0,R1,R2}), (R4,0,{R0,R1,R2}) }
           ^^               ^^

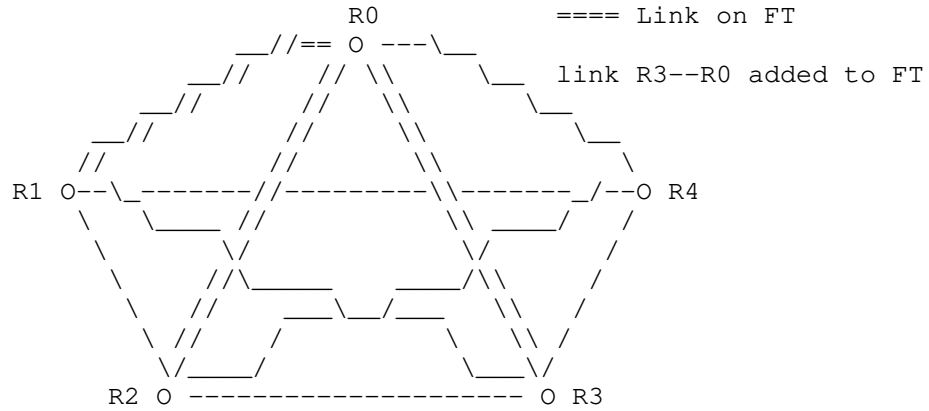
```



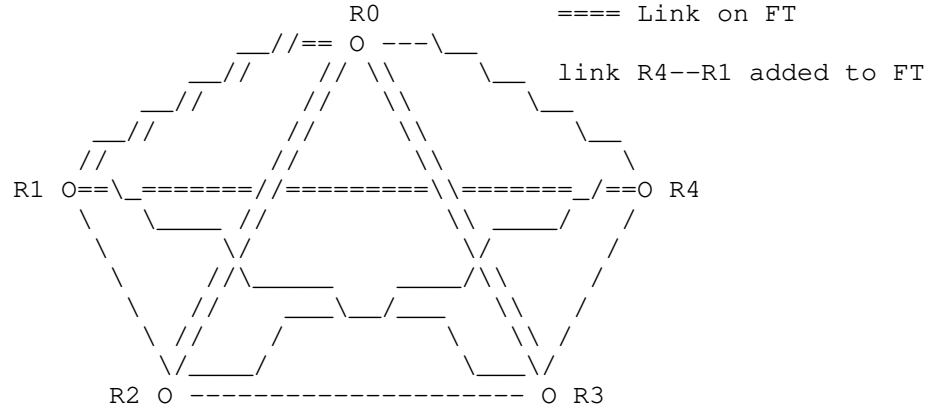
```

3. //remove the 1st element (R3,0,{R0,R1,R2}) from Cq, R0's D=2 < MaxD
   Cq = { (R4,0,{R0,R1,R2}) }
   // add (R3,1,{R0}) into FT, increase R0's D by one
   FT = { (R0,3,{  }), (R1,1,{R0}), (R2,1,{R0}), (R3,1,{R0}) }
           ^^^               ^^^^^^^^^^^
   // for Ri connected to R3 (in Cq) not on FT, append R3 to Ri's PHs
   Cq = { (R4,0,{R0,R1,R2,R3}) }.
           ^^

```

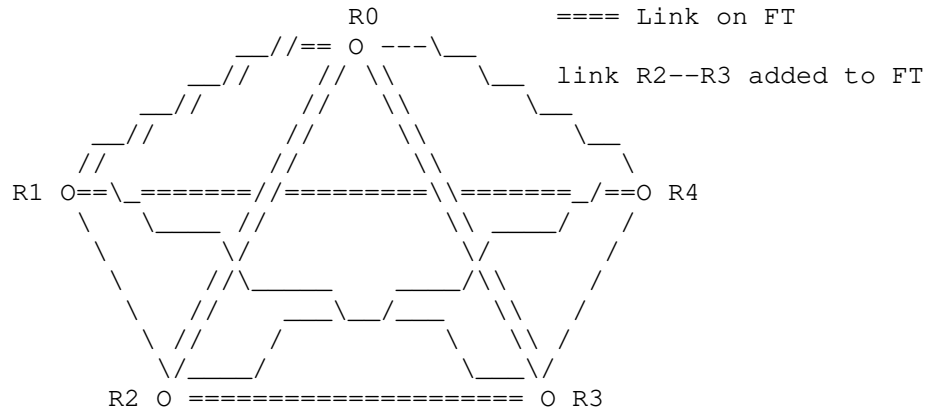


4. //remove the 1st element (R4,0,{R0,R1,R2,R3}) from Cq,R1's D=1 < MaxD  
 Cq = { }  
 // add (R4,1,{R1}) into FT, increase R1's D by one  
 FT = {(R0,3,{})}, (R1,2,{R0}), (R2,1,{R0}), (R3,1,{R0}), (R4,1,{R1})}



All nodes are on FT now. In the following, for each node on FT whose D = 1 (from minimum to maximum ID), link L attached to it and not on FT is found such that L's remote node has minimum D and ID. L is added into FT.

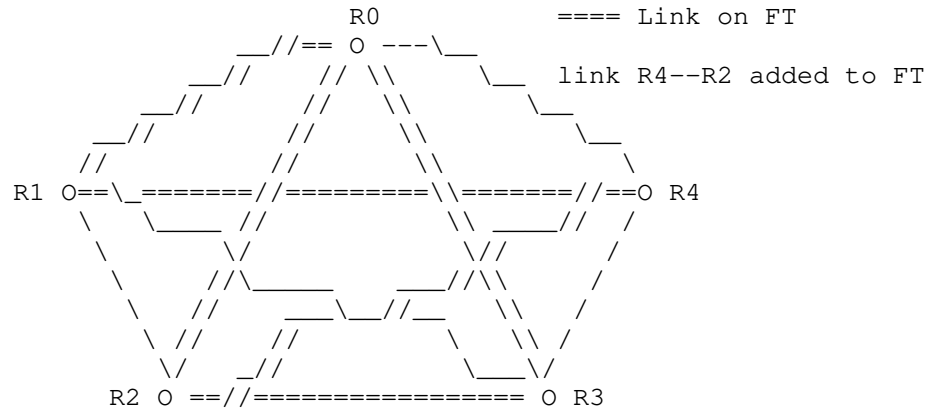
5. // On FT, get node R2 with smallest ID whose D=1  
 FT = {(R0,3,{})}, (R1,2,{R0}), (R2,1,{R0}), (R3,1,{R0}), (R4,1,{R1})}  
 // Add link R2--R3 to FT, ^^^^^^^^^^^  
 // where R2--R3 is not on FT, R3's D=1 is minimum first and then  
 // R3's ID is minimum (R3 and R4 tie for D), R2's D++ and R3's D++  
 FT = {(R0,3,{})}, (R1,2,{R0}), (R2,2,{R0,R3}), (R3,2,{R0}), (R4,1,{R1})}



```

6. // On FT, get node R4 with smallest ID whose D=1
   FT = {(R0,3,{ }),(R1,2,{R0}),(R2,2,{R0,R3}),(R3,2,{R0}),(R4,1,{R1})}
   // Add link R4--R2 to FT, where
   // R4--R2 is not on FT, R2's D=2 is minimum first and then R2's ID is
   // minimum (R2 and R3 tie for D), increase R2's D and R4's D by one
   FT = {(R0,3,{ }),(R1,2,{R0}),(R2,3,{R0,R3}),(R3,2,{R0}),(R4,2,{R1,R2})}

```



FT is computed, which has Degree of 3 and Diameter of 2.

#### Authors' Addresses

Huaimo Chen  
Futurewei  
Boston  
USA

Email: [huaimo.chen@futurewei.com](mailto:huaimo.chen@futurewei.com)

Mehmet Toy  
Verizon  
USA

Email: [mehmet.toy@verizon.com](mailto:mehmet.toy@verizon.com)

Yi Yang  
IBM  
Cary, NC  
United States of America

Email: [yyietf@gmail.com](mailto:yyietf@gmail.com)

Aijun Wang  
China Telecom  
Beiqijia Town, Changping District  
Beijing 102209  
China

Email: wangaj3@chinatelecom.cn

Xufeng Liu  
Volta Networks  
McLean, VA  
USA

Email: xufeng.liu.ietf@gmail.com

Yanhe Fan  
Casa Systems  
USA

Email: yfan@casa-systems.com

Lei Liu  
Fujitsu  
USA

Email: liulei.kddi@gmail.com



LSR Working Group  
Internet-Draft  
Updates: 3563 5305 6232 6233 (if  
approved)  
Intended status: Standards Track  
Expires: October 5, 2019

L. Ginsberg  
P. Wells  
Cisco Systems  
T. Li  
Arista Networks  
T. Przygienda  
S. Hegde  
Juniper Networks, Inc.  
April 3, 2019

Invalid TLV Handling in IS-IS  
draft-ginsberg-lsr-isis-invalid-tlv-03

Abstract

Key to the extensibility of the Intermediate System to Intermediate System (IS-IS) protocol has been the handling of unsupported and/or invalid Type/Length/Value (TLV) tuples. Although there are explicit statements in existing specifications, deployment experience has shown that there are inconsistencies in the behavior when a TLV which is disallowed in a particular Protocol Data Unit (PDU) is received.

This document discusses such cases and makes the correct behavior explicit in order to insure that interoperability is maximized.

This document when approved updates RFC3563, RFC5305, RFC6232, and RFC6233.

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 October 5, 2019.

#### 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. TLV Codepoints Registry . . . . .	3
3. TLV Acceptance in PDUs . . . . .	4
3.1. Handling of Disallowed TLVs in Received PDUs other than LSP Purges . . . . .	4
3.2. Special Handling of Disallowed TLVs in Received LSP Purges . . . . .	4
3.3. Applicability to sub-TLVs . . . . .	5
3.4. Correction to POI TLV Registry Entry . . . . .	5
4. TLV Validation and LSP Acceptance . . . . .	5
5. IANA Considerations . . . . .	6
6. Security Considerations . . . . .	6
7. Acknowledgements . . . . .	6
8. References . . . . .	6
8.1. Normative References . . . . .	6
8.2. Informative References . . . . .	8
Authors' Addresses . . . . .	8

#### 1. Introduction

The Intermediate System to Intermediate System (IS-IS) protocol utilizes Type/Length/Value (TLV) encoding for all content in the body of Protocol Data Units (PDUs). New extensions to the protocol are supported by defining new TLVs. In order to allow protocol

extensions to be deployed in a backwards compatible way an implementation is required to ignore TLVs that it does not understand. This behavior is also applied to sub-TLVs, which are contained within TLVs.

A corollary to ignoring unknown TLVs is having the validation of PDUs be independent from the validation of the TLVs contained in the PDU. PDUs which are valid MUST be accepted even if an individual TLV contained within that PDU is invalid in some way.

These behaviors are specified in existing protocol documents - principally [ISO10589] and [RFC5305]. In addition, the set of TLVs (and sub-TLVs) which are allowed in each PDU type is documented in the TLV Codepoints Registry ( <https://www.iana.org/assignments/isis-tlv-codepoints/isis-tlv-codepoints.xhtml> ) established by [RFC3563] and updated by [RFC6233] and [RFC7356].

This document is intended to clarify some aspects of existing specifications and thereby reduce the occurrence of non-conformant behavior seen in real world deployments. Although behaviors specified in existing protocol specifications are not changed, the clarifications contained in this document serve as updates to RFC 3563 (see Section 2), RFC 5304, and RFC 6233 (see Section 3).

## 2. TLV Codepoints Registry

[RFC3563] established the IANA managed IS-IS TLV Codepoints Registry for recording assigned TLV code points [TLV\_CODEPOINTS]. The initial contents of this registry were based on [RFC3359].

The registry includes a set of columns indicating in which PDU types a given TLV is allowed:

IIH - TLV is allowed in Intermediate System to Intermediate System Hello (IIH) PDUs (Point-to-point and LAN)

LSP - TLV is allowed in Link State PDUs (LSP)

SNP - TLV is allowed in Sequence Number PDUs (SNP) (Partial Sequence Number PDUs (PSNP) and Complete Sequence Number PDUS (CSNP))

Purge - TLV is allowed in LSP Purges [RFC6233]

If "Y" is entered in a column it means the TLV is allowed in the corresponding PDU type.

If "N" is entered in a column it means the TLV is NOT allowed in the corresponding PDU type.

### 3. TLV Acceptance in PDUs

This section describes the correct behavior when a PDU is received which contains a TLV which is specified as disallowed in the TLV Codepoints Registry.

#### 3.1. Handling of Disallowed TLVs in Received PDUs other than LSP Purges

[ISO10589] defines the behavior required when a PDU is received containing a TLV which is "not recognised". It states (see Sections 9.3 - 9.13):

"Any codes in a received PDU that are not recognised shall be ignored."

This is the model to be followed when a TLV is received which is disallowed. Therefore TLVs in a PDU (other than LSP purges) which are disallowed MUST be ignored and MUST NOT cause the PDU itself to be rejected by the receiving IS.

#### 3.2. Special Handling of Disallowed TLVs in Received LSP Purges

When purging LSPs [ISO10589] recommends (but does not require) the body of the LSP (i.e., all TLVs) be removed before generating the purge. LSP purges which have TLVs in the body are accepted though any TLVs which are present "MUST" be ignored.

When cryptographic authentication [RFC5304] was introduced, this looseness when processing received purges had to be addressed in order to prevent attackers from being able to initiate a purge without having access to the authentication key. [RFC5304] therefore imposed strict requirements on what TLVs were allowed in a purge (authentication only) and specified that:

"ISes MUST NOT accept purges that contain TLVs other than the authentication TLV".

This behavior was extended by [RFC6232] which introduced the Purge Originator Identification (POI) TLV and [RFC6233] which added the "Purge" column to the TLV Codepoints registry to identify all the TLVs which are allowed in purges.

The behavior specified in [RFC5304] is not backwards compatible with the behavior defined by [ISO10589] and therefore can only be safely enabled when all nodes support cryptographic authentication. Similarly, the extensions defined by [RFC6233] are not compatible with the behavior defined in [RFC5304], therefore can only be safely enabled when all nodes support the extensions.

It is recommended that implementations provide controls for the enablement of behaviors that are not backward compatible.

### 3.3. Applicability to sub-TLVs

[RFC5305] introduced sub-TLVs, which are TLV tuples advertised within the body of a parent TLV. Registries associated with sub-TLVs are associated with the TLV Codepoints Registry and specify in which TLVs a given sub-TLV is allowed. As with TLVs, it is required that sub-TLVs which are disallowed MUST be ignored on receipt.

### 3.4. Correction to POI TLV Registry Entry

An error was introduced by [RFC6232] when specifying in which PDUs the POI TLV is allowed. Section 3 of [RFC6232] stated:

"The POI TLV SHOULD be found in all purges and MUST NOT be found in LSPs with a non-zero Remaining Lifetime."

However, the IANA section of the same document stated:

"The additional values for this TLV should be IIH:n, LSP:y, SNP:n, and Purge:y. "

The correct setting for "LSP" is "n". This document corrects that error.

## 4. TLV Validation and LSP Acceptance

The correct format of a TLV and its associated sub-TLVs if applicable are defined in the document(s) which introduce each codepoint. The definition SHOULD include what action to take when the format/content of the TLV does not conform to the specification (e.g., "MUST be ignored on receipt"). When making use of the information encoded in a given TLV (or sub-TLV) receiving nodes MUST verify that the TLV conforms to the standard definition. This includes cases where the length of a TLV/sub-TLV is incorrect and/or cases where the value field does not conform to the defined restrictions.

However, the unit of flooding for the IS-IS Update process is an LSP. The presence of a TLV (or sub-TLV) with content which does not conform to the relevant specification MUST NOT cause the LSP itself to be rejected. Failure to follow this requirement will result in inconsistent LSP Databases on different nodes in the network which will compromise the correct operation of the protocol.

LSP Acceptance rules are specified in [ISO10589] . Acceptance rules for LSP purges are extended by [RFC5304] [RFC5310] and further extended by [RFC6233].

[ISO10589] also specifies the behavior when an LSP is not accepted. This behavior is NOT altered by extensions to the LSP Acceptance rules i.e., regardless of the reason for the rejection of an LSP the Update process on the receiving router takes the same action.

## 5. IANA Considerations

IANA is requested to update the TLV Codepoints Registry to reference this document.

IANA is also requested to modify the entry for the POI TLV in the TLV Codepoints Registry to be:

IIH:n, LSP:n, SNP:n, and Purge:y.

## 6. Security Considerations

As this document makes no changes to the protocol there are no new security issues introduced.

The clarifications discussed in this document are intended to make it less likely that implementations will incorrectly process received LSPs, thereby also making it less likely that a bad actor could exploit a faulty implementaion.

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

## 7. Acknowledgements

The authors would like to thank Alvaro Retana.

## 8. References

### 8.1. 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>>.
- [RFC3563] Zinin, A., "Cooperative Agreement Between the ISOC/IETF and ISO/IEC Joint Technical Committee 1/Sub Committee 6 (JTC1/SC6) on IS-IS Routing Protocol Development", RFC 3563, DOI 10.17487/RFC3563, July 2003, <<https://www.rfc-editor.org/info/rfc3563>>.
- [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>>.
- [RFC6232] Wei, F., Qin, Y., Li, Z., Li, T., and J. Dong, "Purge Originator Identification TLV for IS-IS", RFC 6232, DOI 10.17487/RFC6232, May 2011, <<https://www.rfc-editor.org/info/rfc6232>>.
- [RFC6233] Li, T. and L. Ginsberg, "IS-IS Registry Extension for Purges", RFC 6233, DOI 10.17487/RFC6233, May 2011, <<https://www.rfc-editor.org/info/rfc6233>>.
- [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>>.
- [TLV\_CODEPOINTS] IANA, "IS-IS TLV Codepoints web page (<https://www.iana.org/assignments/isis-tlv-codepoints/isis-tlv-codepoints.xhtml>)".

## 8.2. Informative References

[RFC3359] Przygienda, T., "Reserved Type, Length and Value (TLV) Codepoints in Intermediate System to Intermediate System", RFC 3359, DOI 10.17487/RFC3359, August 2002, <<https://www.rfc-editor.org/info/rfc3359>>.

## Authors' Addresses

Les Ginsberg  
Cisco Systems

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

Paul Wells  
Cisco Systems

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

Tony Li  
Arista Networks  
5453 Great America Parkway  
Santa Clara, California 95054  
USA

Email: [tony.li@tony.li](mailto:tony.li@tony.li)

Tony Przygienda  
Juniper Networks, Inc.  
1194 N. Matilda Ave  
Sunnyvale, California 94089  
USA

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

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

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



LSR Working Group  
Internet-Draft  
Intended status: Informational  
Expires: August 15, 2019

S. Hares  
Huawei  
February 11, 2019

IPRAN Grid-Ring IGP convergence problems  
draft-hares-lsr-grid-ring-convergence-00.txt

Abstract

This draft describes problems with IGP convergence time in some IPRAN networks that use a physical topology of grid backbones that connect rings of routers. Part of these IPRAN network topologies exist in data centers with sufficient power and interconnections, but some network equipment sits in remote sites impacted by power loss. In some geographic areas these remote sites may be subject to rolling blackouts. These rolling power blackouts could cause multiple simultaneous node and link failures. In these remote networks with blackouts, it is often critical that the IPRAN phone network re-converge quickly.

The IGP running in these networks may run in a single level of the IGP. This document seeks to briefly describe these problems to determine if the emerging IGP technologies (flexible algorithms, dynamic flooding, layers of hierarchy in IGPs) can be applied to help reduce convergence times. It also seeks to determine if the improvements of these algorithms or the IP-Fast re-route algorithms are thwarted by the failure of multiple link and nodes.

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 August 15, 2019.

## 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. IPRAN Topologies . . . . .	3
3. Definitions . . . . .	7
3.1. Requirements language . . . . .	7
4. Problem detection using theoretical IGP Convergence . . . . .	8
4.1. Equation applied to Data Center IGP Convergence . . . . .	9
4.2. Flooding Problem on the Rings . . . . .	11
4.3. Flooding problem on the grid . . . . .	12
5. Multiple simultaneous link and node failures . . . . .	12
5.1. Multiple link failures on Ring . . . . .	13
5.2. Multiple link failures on Grid . . . . .	14
6. Problem with Flat ISIS areas . . . . .	14
7. Problems with Dense Flooding Algorithm . . . . .	15
8. References . . . . .	15
8.1. Normative References: . . . . .	15
8.2. Informative References . . . . .	15
Author's Address . . . . .	17

## 1. Introduction

This draft describes problems with IGP convergence time in some IPRAN networks. The physical topologies of these IPRAN networks combine a grid backbone topology with a ring topology to support phone networks (see figure 1). Routers are attached to the rings that route traffic from the IPRAN devices (see figure 2). Each of the rings is attached to two grid nodes in order to provide redundancy. All of the routers in the IPRAN ring-grid network topology run a single IGP (IS-IS).

Some current deployments attach 10-30 routers per ring with a 20 by 20 grid of routers. In these deployments, a grid of 400 routers supports between 10,000 - 15,000 routers on the IPRAN rings.

Convergence of the IGP after a single link failure on one ring router is over 1 second for these topologies. The desired convergence time for a single link failure is less than 200 ms for phone networks.

Initial convergence of the full network may take on the order of minutes.

Part of these IPRAN network topologies exist in data centers with sufficient power and interconnections, but some network equipment sits in remote sites impacted by power loss. In some geographic regions, these remote sites may be subject to rolling blackouts. These rolling power blackouts could cause multiple simultaneous link or node failures. In these remote networks with blackouts, it is often critical that the IPRAN network converge quickly to restore what mobile phone service it can. Keeping isolated portions of the network working may be critical to keep some phone service working. Converging the isolated portions back into the network when repairs are made also causes further disruptions.

Due to the topologies of the IPRAN network, this document examines how the flooding of IGP informations causes the longer IGP convergence times for single links. The potential multiple simultaneous link and node failures mean that the assumptions in most IGP and fast IP-Route algorithms do not apply.

This document seeks to briefly describe these problems to determine if the following emerging IGP technologies can be applied to solve the convergence problem:

- flexible algorithms [I-D.ietf-lsr-flex-algo],
- dynamic flooding [I-D.li-lsr-dynamic-flooding],
- Level 1 abstraction for ISIS [I-D.li-area-abstraction]
- hierarchical IS-IS [I-D.li-hierarchical-isis]

## 2. IPRAN Topologies

A bit of background on the IPRan sizes.

Grid topologies can be any size of square topologies. Figure 1 shows a 3 router by 3 router topologies (3x3) with 9 nodes). Other sizes could be 10 routers by 10 routers (10X10) with 100 nodes, 15 routers by 15 routers (15X15) with 225 routers, or 50 nodes by 50 nodes (20X20) with 400 routers. A grid with network topology of a 100x100 grid would have 10,000 grid-routers (grid only and ring-grid). Suppose that for every two grid nodes, 3 rings would be attached and

on each ring there are 50 nodes. This topology would result in 750,000 ring routers plus 10,000 grid routers. The size of this topology rivals data center sizes, but the IPRAN network does not have the infrastructure advantages of the data center.

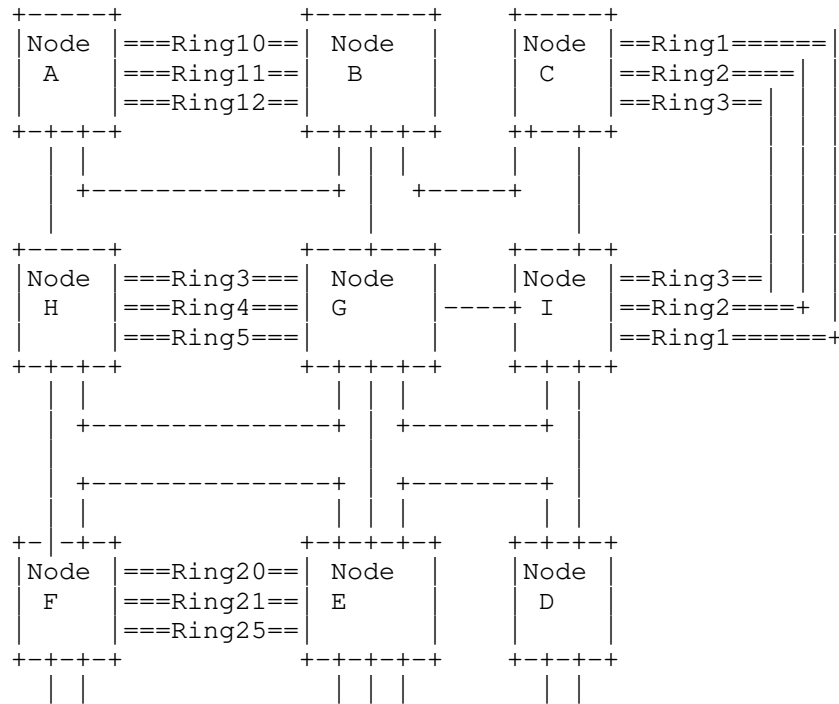


Figure 1

Figure 1: Example IPRAN Grid-Ring Topology

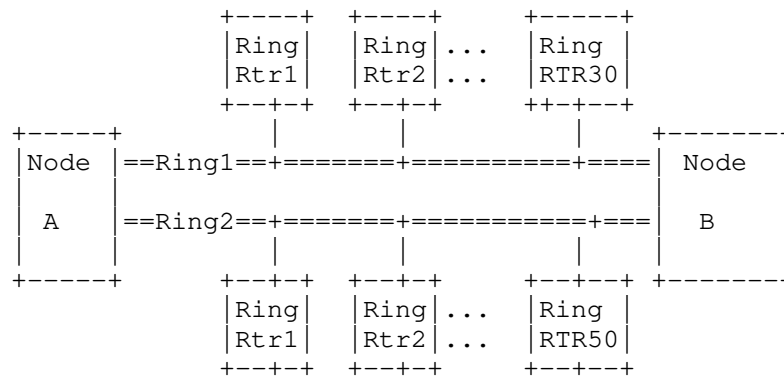


Figure 2

Figure 2: Example IPRAN Ring Topology

One characteristics of a grid is that a basic 3X3 square can be overlaid on most grids. Figure 3 shows a 10 by 10 grid with 3 by 3. Notice that the grid squares overlaid on column 10 and row 10 form partial squares (see GS4, GS8, GS12, GS13, GS14, GS15, and GS16).

If additional connections were made most of column 10 could form a single Grid (GS4, GS8, and GS12), and most of row 10 could form a single grid (GS13, GS14, and GS15). Alternatively, with a single connection, GS16 could merge with GS15 to form a partial grid of 4 nodes.

X = Grid node  
GS = Grid Square 1

GS1	GS2	GS3	GS4
X X X	X X X	X X X	X
X X X	X X X	X X X	X
X X X	X X X	X X X	X
GS5	GS6	GS7	GS8
X X X	X X X	X X X	X
X X X	X X X	X X X	X
X X X	X X X	X X X	X
GS9	GS10	GS11	GS12
X X X	X X X	X X X	X
X X X	X X X	X X X	X
X X X	X X X	X X X	X
GS13	GS14	GS15	GS16
X X X	X X X	X X X	X

Figure 3

Figure 3: Overlaying Grid Squares on IPRAN Grid

The grid topology is currently one flat IGP. However, logical grid squares could form Level 1 areas within the IGP. If one desired to create an L1 Area abstraction such as defined [I-D.li-area-abstraction], then the grid-square areas could be created as L1 areas and connected by 1-3 links to adjacent areas. Figure 4 shows a logical topology for grid squares 1-8 from figure 2.

X = Grid node  
 G = Grid node G, Area Leader  
 GS<sub>n</sub> = Grid Square n (1-8)  
 Layer 2 area (1-8)

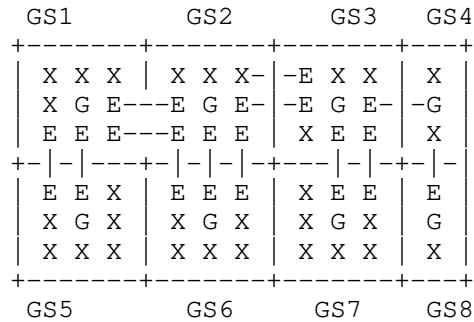


Figure 4

Figure 4: Grid Squares Area Leaders and Area Edge Nodes

### 3. Definitions

This section provides definitions for nodes within the IPRAN routing infrastructure:

ring router: a routing device only attach to a ring in an IPRAN topology which routes end-system information

ring-grid router routing device attached to ring and the grid topology

grid router: a routing device which is only attached to the IPRAN Grid network

pseudo-node for grid area: a pseudo-node which summarizes for an IGP a grid area at one level for a higher level.

#### 3.1. 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].

#### 4. Problem detection using theoretical IGP Convergence

Theoretical "best" convergence times for a single link failure on ring depths of 30 nodes suggests the flooding time is a major component for the flat IGP. Estimates of theoretical best convergence times may be based on set of equations shown in figure 5. These equations show how network convergence is the maximum time for the information on a link change (down (failure) or up) to spread to all routers in the network. The change travels along a pathway of routers from the change to any particular router. Therefore, convergence is really topology dependent on the convergence time in each router and the pathways.

The theoretical convergence equations in figure 5 include updating the RIB/FIB (Trib) and forwarding elements (Tdd). Some IGPS may forward IGP traffic after calculating the SPF (Tspf) and updating the RIB/FIB, but before updating the FIB line cards (Tdd). In this case, these factors would be zero in the equation.

If several factors are zero or a constant, then the convergence may be determined by one element in the equation that dominates the convergence per node.



```

CT-Node = Td + To + Tf + Tspf + Trib + Tdd

CT-Node = Node convergence time
Td = link failure detection time
    (or link up detection time)
To = time to originate LSP
    describing the new topology

Tf = Time to flood the change
    from this node to other nodes
    that must perform a flood update

Tspf = Time for shortest path calculation

Trib = Time to update the RIB and FIB

Tdd = time to distribute the FIB to line cards

CT-path(i) = sum [CT-Node(j), .. CT-Node-(n)]
              where i =path through network
              j = nodes on path (1..n)

CTnetwork = maximum (CT-path(i))
              where i = 0..n paths
Figure 5

```

Figure 5: Convergence equations

[My first experience with an equation like this was Cengiz Alaettinoglu research in IGP around 2000 at NANOG. (Please let me know if you have a good scholarly reference or presentation reference for these equations).]

#### 4.1. Equation applied to Data Center IGP Convergence

Some early SPF implementations were slow with large IGP topologies. In this case, IGP's SPF calculations dominates the convergence time for all nodes. Thus the Tspf dominates the time for each network path and the entire networks convergence time. One might summarize the convergence as:

CT-network = (Tspf + constant) \* maximum path-length

The maximum path length is often called the network depth. The network depth of a full mesh network is 1. The network depth of a dense mesh fat tree in a data center with 3 levels (top of rack, aggregate, spine) is 3. If Tspf dominates the calculation then:

$$CT\text{-}network = (Tspf + constant) * 3$$

Centralized algorithms might improve convergence time if Tspf is the main factor. Rather than using routers with typically low calculation power, centralized devices could be optimized for the calculation. If the difference in network depth of sending the information end-to-end on any network path and sending it to the centralized processor and back is minimal, then centralized processing may be more effective.

If flooding (Tf) dominates the per node convergence, the equation is:

$$CT\text{-}network = (Tf + constant) * 3$$

Many of the authors of the IGP flooding enhancements to reduce the data flooded understand that the flooding depends on the maximum pathway length for pathways in the IGP graph. (see 802.1aq [I-D.allan-lsr-flooding-algorithm], Li et al. [I-D.li-lsr-dynamic-flooding], Shen, Ginsberg, and Thyamagundalu [I-D.shen-isis-spine-leaf-ext]). Others mention creating a sub-graph of the entire topology to reduce the flooding traffic and reduce convergence time (Chen et al. [I-D.cc-ospf-flooding-reduction]).

Some of the IGP flooding reductions are identifying and limiting the number of global pathways without mentioning their concern for length. (see Chunduri and Eckert [I-D.ce-lsr-ppr-graph]).

The point behind this is that each algorithm has a set of goals. Those goals may impact other things that impact convergence. Some questions one can ask are:

- o Does the algorithm seek to reduced data flooded and stored?
- o Does the algorithm seek to reduce convergence time?
- o If the algorithm tries to both reduce the data flooded and stored, what trade-offs did the algorithm make?
- o what is the impact of the topology?

If one looks to adapt the algorithms developed for the dense interconnections of the 3 tier data center to the IPRAN Grid-ring network structure, these questions are important.

#### 4.2. Flooding Problem on the Rings

Putting 30 or 50 ring routers on a ring may help operational costs. Within a city the higher density of rings may allow more cells for the phone. In the rural networks, it may allow the cells to be deployed over a larger physical area.

Every router one puts on a ring increases the network depth of the path through a fully operational ring or a partitioned ring that is still connected to the network. The network depth of a ring is

$$\text{network depth} = (n\text{-ring-nodes} + n\text{-grid-ring})/2$$

where

$$n\text{-ring-nodes} = 30 \text{ to } 50 \text{ nodes}$$

$$n\text{-grid-nodes} = 2 \text{ nodes}$$

A partitioned ring may have the full network depth if the link between a grid-router and the ring router attached to it fails.

This flooding time is only for the on-ring path. For a network path that involves the link failure of a ring router link the pathway is:

$$\begin{aligned} \text{network depth} = & \text{depth(failed-ring)} + \\ & \text{depth(grid)} + \\ & \text{depth(remote-ring)} \end{aligned}$$

$$\text{depth(failed-ring)} = \text{network depth of ring with failed link.}$$

$$\text{depth(grid)} = \text{network depth of pathway through Grid}$$

$$\text{depth(remote-ring)} = \text{network depth of pathway through remote ring}$$

Figure 6

Figure 6: Convergence equations

The worse case IGP convergence time combines the worse case for each of these network depths.

#### 4.3. Flooding problem on the grid

The network depth of grid topologies grows as the size of the grid grows from 3X3 to 10X10 to 100X100. The network depth of the best case pathway through the grid is a single hop as it is on the same ring-grid router. The worse case path is the one from x1 to X2 in figure 7. A network pathway that goes from x1 to X2 by using routers in the following grid squares: pathway of GS2, GS3, GS4, GS8, GS12 could take 19 hops.

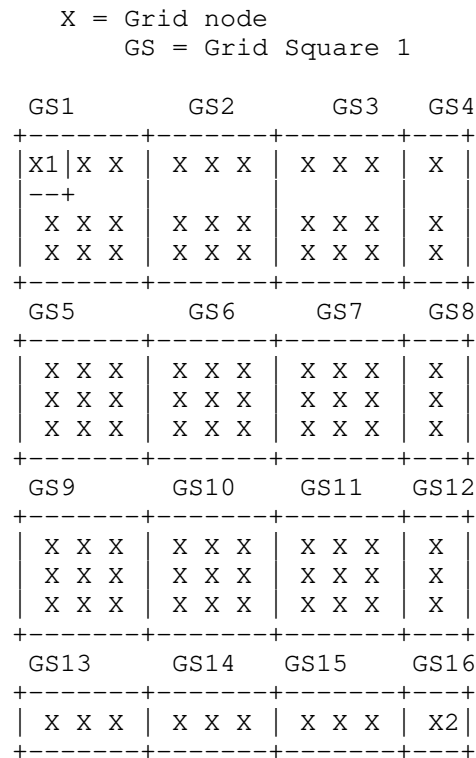


Figure 6

Figure 7: Worse Case for 10X10 Grid

#### 5. Multiple simultaneous link and node failures

Part of these IPRAN network topologies exist in data centers with power and connective, but some do not. Ring routers are more likely

to be at remote sites where power loss can occur. However, some ring-grid routers or grid-only routers may be in remote sites.

In some geographic locations, power losses can be rolling blackouts that cause multiple link and node outages during the failure. These outages may be unpredictable due to weather or natural disasters, or semi-predictable due to brownouts. Upon attempts to restore power, the restorations may have mixed combinations of links and nodes up. Multiple simultaneous link and node failures may impact both the ring topologies and the grid topologies in the IPRAN network.

For simplicity of this discussion, I will present the node outages as the outages of all links. A node outage may take far longer if rebooting the routers or reconfiguring spare ring routers takes a long time. For this initial pass on this document, I will simply treat node outages as failure of all links for a time period that clear all valid paths.

Most fast re-route technology such LFA [RFC5286] or MRT [RFC7812] set-up IP backup paths to route around a single link or node failure. In fact, the MRT architecture explicitly states that

"MRT-FRR creates two alternative forwarding trees that ... are maximally diverse from one another, providing link and node protect for 100% of paths and failures as long as the failures do not cut the network into multiple pieces"

#### 5.1. Multiple link failures on Ring

Ring routers may be located at sites that may lose connection to the ring or to a grid-ring router. A single link failure may cut the ring, but leave all nodes attached if the failed link is between one of the ring routers (single on ring) or between the a ring-grid routers and a ring router.

Multiple link failures on a ring will cause the ring to partition, isolating some nodes. One way to handle this is to ignore the convergence on the partitioned rings. Since local phone service during these outages may be useful, it may be important for the IGP's on the isolated portions of the rings to continue to operate. During the restoration phase, additional links may appear to go up and down as the partitions heal. Several isolated portions of the ring may be restored to form a larger isolated portion of the ring. Eventually, the isolated parts should reconnect to a fully connected ring.

## 5.2. Multiple link failures on Grid

Multiple link failures can occur on the ring-grid routers or grid-only routers. These failures may dramatically impact the data forwarding pathways through the grid and the flooding pathways. Fast convergence of the grid depends on an algorithm tuned for the grid topologies.

The failures on the grid can impact different parts of the IGP convergence algorithm.

## 6. Problem with Flat ISIS areas

Abstraction in an IGP can provide a logical means to scale IGPs. Creating 2 levels of topology in the IPRAN network based on ISIS areas could reduce the network depth and the size of the topology database in level devices.

However, as Li states in [I-D.li-area-abstraction] the ISIS concepts work well if:

- o "the Level 1 area is tangential to the Level 2 area", or
- o if "there are a number of routers in both level 1 and level 2 and they are adjacent".

However it does not work well if Level 1 area needs to provide transit for level 2 traffic.

Suppose all ring routers networks were placed in level 1 areas, and grid-only routers were in level 2. The ring-grid routers are in both level 1 and 2. This reduces the current topology to a topology similar to the spine-leaf topology. While this reduces the amount of LSP stored, it may not significantly improve IGP convergence. The flooding topology must be examined to determine the maximum network depth, and the router operations must be examined to determine the per IGP flooding time.

It also restricts repair of an L2 Grid path via a L1 Ring. This repair might be necessary in the multi-failure scenario.

The area abstraction described in [I-D.li-area-abstraction] could be used to remove these restrictions.

Additional levels of hierarchy described by Li in [I-D.li-hierarchical-isis] could be utilized in the grid to allow additional levels of abstractions. These levels could reduce the network depth that IGP flooding passes through.

One difficulty with using abstraction provided by areas and levels is the configuration of the appropriate network topology with multiple levels, and reconfigurations of these levels. To be effective for 100X100 grids, it would be beneficial to automate the configuration of areas.

## 7. Problems with Dense Flooding Algorithm

- o spine-leaves - rings may be leaves, but grid is not spine-leave topology.
- o sparse link flooding - Grid may have too little or too much. Top priority is fast convergence not reduced load of LSPF, but fast convergence.
- o preferred path graph - goal is preferred path reduction of the number of preferred paths through network. Fast re-route also sets up paths. The preferred path graph needs to be carefully integrated with any fast reroute scheme.
- o flooding of 802.1aq - is designed for dense mesh.
  - \* The algorithm's two tree structure of 802.1aq provide complete coverage in the presence of a single link failure while constraining the number of LSAs.
  - \* Both trees in the two structure have the same convergence properties in the IPRAN ring and grid.

## 8. References

### 8.1. Normative References:

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

### 8.2. Informative References

- [I-D.allan-lsr-flooding-algorithm]  
Allan, D., "A Distributed Algorithm for Constrained Flooding of IGP Advertisements", draft-allan-lsr-flooding-algorithm-00 (work in progress), October 2018.

- [I-D.cc-ospf-flooding-reduction]  
Chen, H., Cheng, D., Toy, M., and Y. Yang, "LS Flooding Reduction", draft-cc-ospf-flooding-reduction-04 (work in progress), September 2018.
- [I-D.ce-lsr-ppr-graph]  
Chunduri, U. and T. Eckert, "Preferred Path Route Graph Structure", draft-ce-lsr-ppr-graph-01 (work in progress), October 2018.
- [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-01 (work in progress), November 2018.
- [I-D.li-area-abstraction]  
Li, T., "Level 1 Area Abstraction for IS-IS", draft-li-area-abstraction-00 (work in progress), June 2018.
- [I-D.li-hierarchical-isis]  
Li, T., "Hierarchical IS-IS", draft-li-hierarchical-isis-00 (work in progress), June 2018.
- [I-D.li-lsr-dynamic-flooding]  
Li, T., Psenak, P., Ginsberg, L., Przygienda, T., Cooper, D., Jalil, L., and S. Dontula, "Dynamic Flooding on Dense Graphs", draft-li-lsr-dynamic-flooding-02 (work in progress), December 2018.
- [I-D.shen-isis-spine-leaf-ext]  
Shen, N., Ginsberg, L., and S. Thyamagundalu, "IS-IS Routing for Spine-Leaf Topology", draft-shen-isis-spine-leaf-ext-07 (work in progress), October 2018.
- [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>>.
- [RFC7812] Atlas, A., Bowers, C., and G. Enyedi, "An Architecture for IP/LDP Fast Reroute Using Maximally Redundant Trees (MRT-FRR)", RFC 7812, DOI 10.17487/RFC7812, June 2016, <<https://www.rfc-editor.org/info/rfc7812>>.



Author's Address

Susan Hares  
Huawei  
Saline  
US

Email: [shares@endzh.com](mailto:shares@endzh.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:

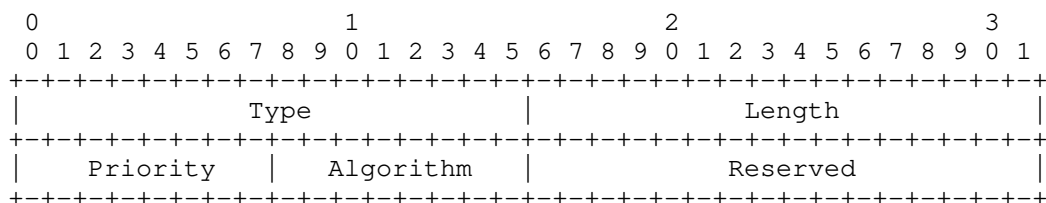
1. A TLV that is used to advertise the preference for becoming Area Leader.
2. A TLV that is used to indicate the support for Dynamic Flooding and the algorithms that the advertising node supports for distributed mode, if any.
3. OSPFv2 Opaque LSA and OSPFv3 LSA to advertise the flooding topology for 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

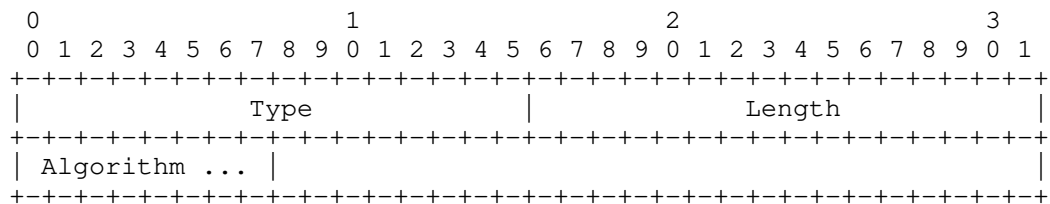
Algorithm: as defined in Section 5.1.1.

### 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:

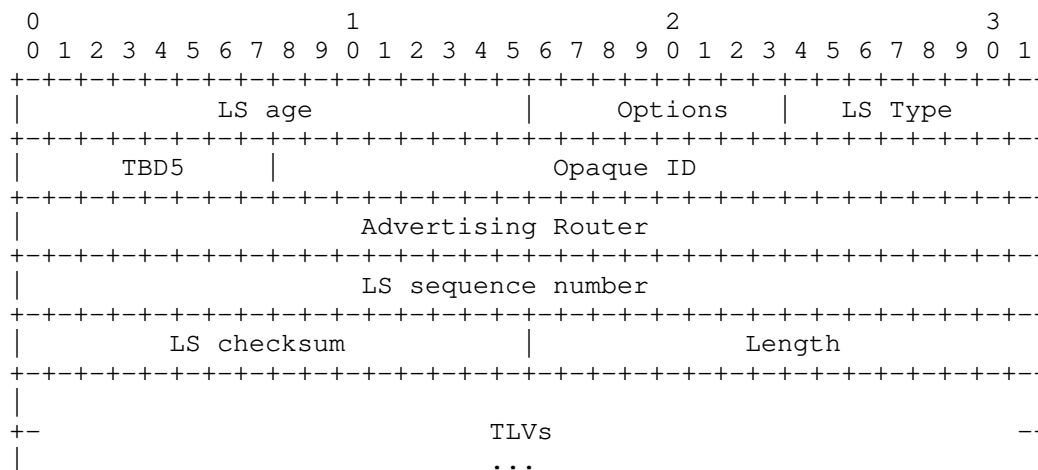


Figure 1: OSPFv2 Dynamic Flooding Opaque LSA

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:

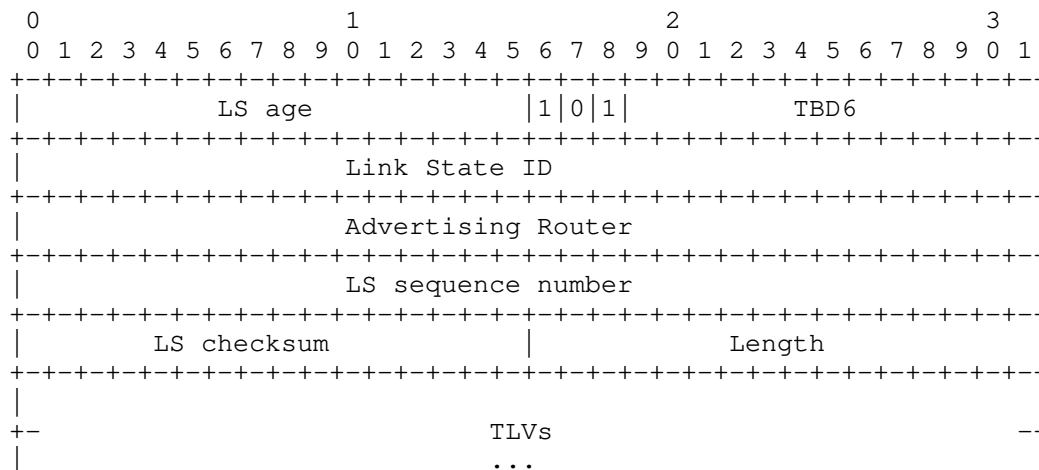


Figure 2: OSPFv3 Dynamic Flooding LSA

#### 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:

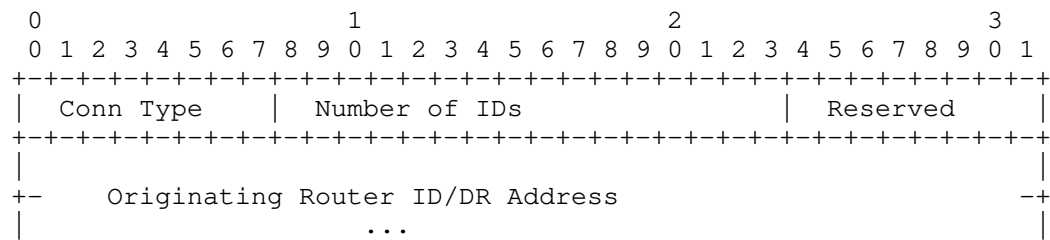


Figure 3: OSPFv2 Router IDs TLV Entry

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:

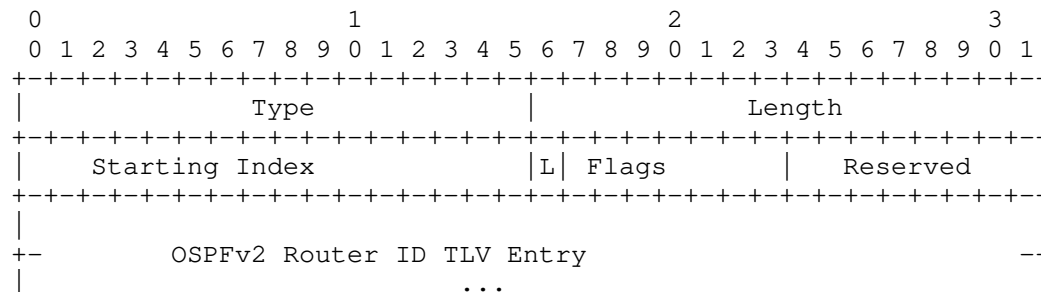


Figure 4: OSPFv2 Area Router IDs TLV

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.

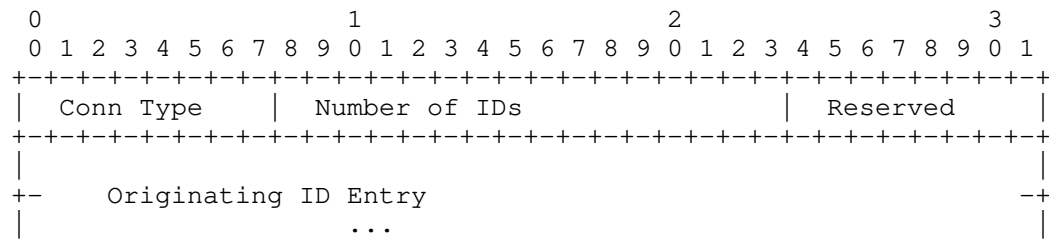
If there are multiple OSPFv2 Area Router ID TLVs with the L bit set advertised by the same router, the TLV which specifies the smaller maximum index is used and the other TLV(s) with L bit set are ignored. TLVs which specify Router IDs with indices greater than that specified by the TLV with the L bit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

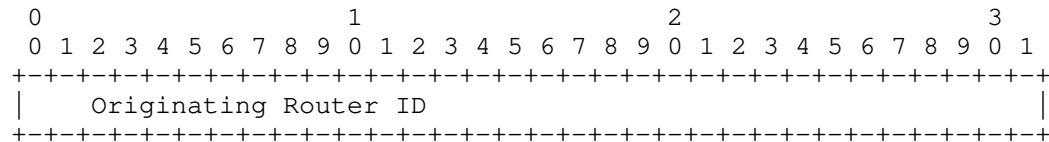
Number of IDs - 2 bytes

Reserved - 1 byte

MUST be transmitted as 0 and MUST be ignored on receipt

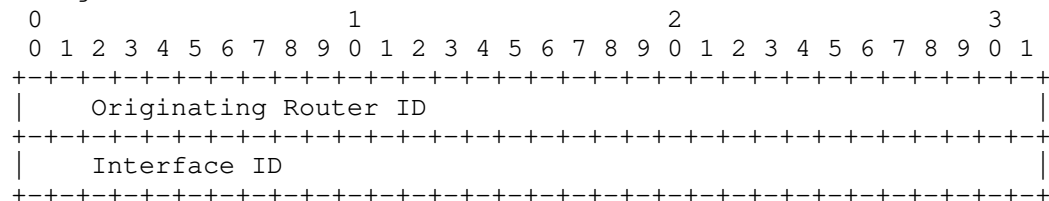
Originating ID Entry takes one of the following forms:

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:

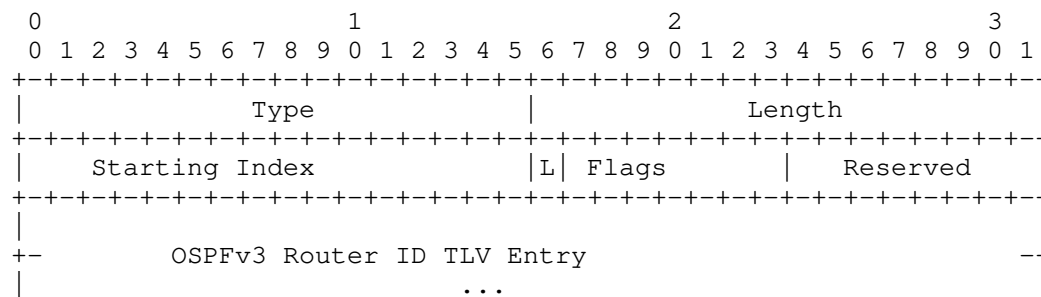


Figure 6: OSPFv3 Area Router IDs TLV

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:

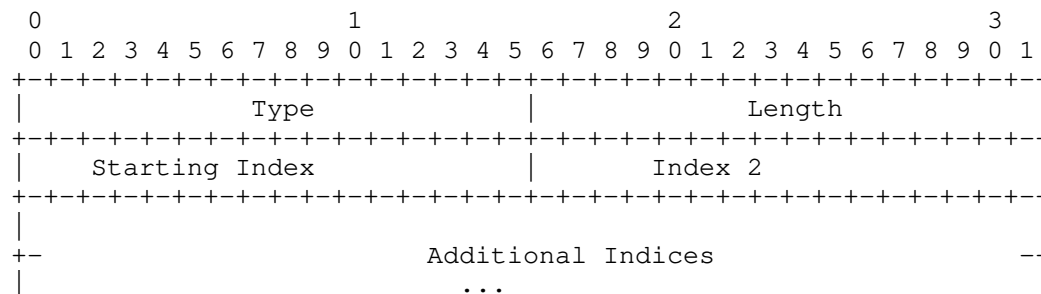


Figure 7: OSPF Flooding Path TLV

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.

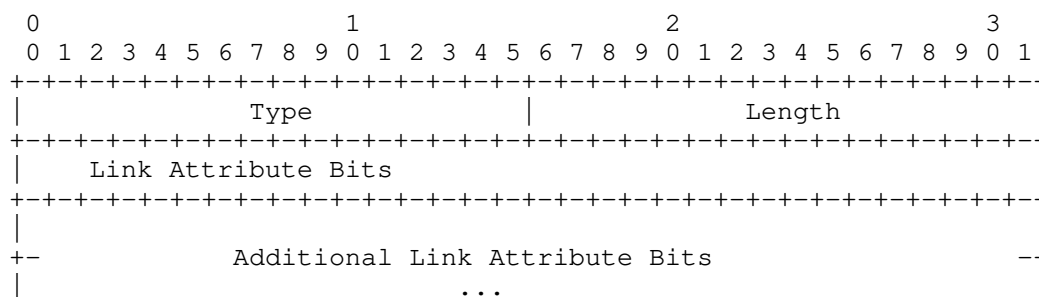


Figure 8: OSPF Link Attributes Bits TLV

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](mailto:tony.li@tony.li)

Tony Przygienda  
Juniper Networks  
1133 Innovation Way  
Sunnyvale, California 94089  
United States of America

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

Peter Psenak (editor)  
Cisco Systems, Inc.  
Eurovea Centre, Central 3  
Pribinova Street 10  
81109 Bratislava  
Slovakia

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

Les Ginsberg  
Cisco Systems, Inc.  
510 McCarthy Blvd.  
Milpitas, California 95035  
United States of America

Email: [ginsberg@cisco.com](mailto: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



Networking Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: March 5, 2020

N. Shen  
L. Ginsberg  
Cisco Systems  
S. Thyamagundalu  
September 2, 2019

IS-IS Routing for Spine-Leaf Topology  
draft-ietf-lsr-isis-spine-leaf-ext-02

Abstract

This document describes a mechanism for routers and switches in a Spine-Leaf type topology to have non-reciprocal Intermediate System to Intermediate System (IS-IS) routing relationships between the leafs and spines. The leaf nodes do not need to have the topology information of other nodes and exact prefixes in the network. This extension also has application in the Internet of Things (IoT).

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 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. Motivations . . . . .	3
3. Spine-Leaf (SL) Extension . . . . .	4
3.1. Topology Examples . . . . .	4
3.2. Applicability Statement . . . . .	5
3.3. Spine-Leaf TLVs . . . . .	6
3.3.1. Spine-Leaf TLV . . . . .	6
3.3.2. Leaf-Set TLV . . . . .	7
3.3.2.1. Leaf-Set Sub-TLVs . . . . .	7
3.3.3. Advertising IPv4/IPv6 Reachability . . . . .	8
3.3.4. Advertising Connection to RF-Leaf Node . . . . .	8
3.4. Mechanism . . . . .	9
3.4.1. Pure CLOS Topology . . . . .	10
3.5. Implementation and Operation . . . . .	11
3.5.1. CSNP PDU . . . . .	11
3.5.2. Leaf to Leaf connection . . . . .	12
3.5.2.1. Local traffic only . . . . .	12
3.5.2.2. Transit traffic allowed . . . . .	12
3.5.3. Spine Node Hostname . . . . .	13
3.5.4. IS-IS Reverse Metric . . . . .	13
3.5.5. Spine-Leaf Traffic Engineering . . . . .	13
3.5.6. Other End-to-End Services . . . . .	13
3.5.7. Address Family and Topology . . . . .	14
3.5.8. Migration . . . . .	14
4. IANA Considerations . . . . .	14
5. Security Considerations . . . . .	15
6. Acknowledgments . . . . .	15
7. References . . . . .	15
7.1. Normative References . . . . .	15
7.2. Informative References . . . . .	17
Authors' Addresses . . . . .	17

## 1. Introduction

The IS-IS routing protocol defined by [ISO10589] has been widely deployed in provider networks, data centers and enterprise campus environments. In the data center and enterprise switching networks, a Spine-Leaf topology is commonly used. This document describes a mechanism where IS-IS routing can be optimized for a Spine-Leaf topology.

In a Spine-Leaf topology, normally a leaf node connects to a number of spine nodes. Data traffic going from one leaf node to another leaf node needs to pass through one of the spine nodes. Also, the decision to choose one of the spine nodes is usually part of equal cost multi-path (ECMP) load sharing. The spine nodes can be considered as gateway devices to reach destinations on other leaf nodes. In this type of topology, the spine nodes have to know the topology and routing information of the entire network, but the leaf nodes only need to know how to reach the gateway devices to which are the spine nodes they are uplinked.

This document describes the IS-IS Spine-Leaf extension that allows the spine nodes to have all the topology and routing information, while keeping the leaf nodes free of topology information other than the default gateway routing information. The leaf nodes do not even need to run a Shortest Path First (SPF) calculation since they have no topology information.

### 1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

## 2. Motivations

- o The leaf nodes in a Spine-Leaf topology do not require complete topology and routing information of the entire domain since their forwarding decision is to use ECMP with spine nodes as default gateways
- o The spine nodes in a Spine-Leaf topology are richly connected to leaf nodes, which introduces significant flooding duplication if they flood all Link State PDUs (LSPs) to all the leaf nodes. It saves both spine and leaf nodes' CPU and link bandwidth resources if flooding is blocked to leaf nodes. For small Top of the Rack (ToR) leaf switches in data centers, it is meaningful to prevent full topology routing information and massive database flooding through those devices.
- o When a spine node advertises a topology change, every leaf node connected to it will flood the update to all the other spine nodes, and those spine nodes will further flood them to all the leaf nodes, causing a  $O(n^2)$  flooding storm which is largely redundant.
- o Similar to some of the overlay technologies which are popular in data centers, the edge devices (leaf nodes) may not need to

contain all the routing and forwarding information on the device's control and forwarding planes. "Conversational Learning" can be utilized to get the specific routing and forwarding information in the case of pure CLOS topology and in the events of link and node down.

- o Small devices and appliances of Internet of Things (IoT) can be considered as leafs in the routing topology sense. They have CPU and memory constrains in design, and those IoT devices do not have to know the exact network topology and prefixes as long as there are ways to reach the cloud servers or other devices.

### 3. Spine-Leaf (SL) Extension

#### 3.1. Topology Examples

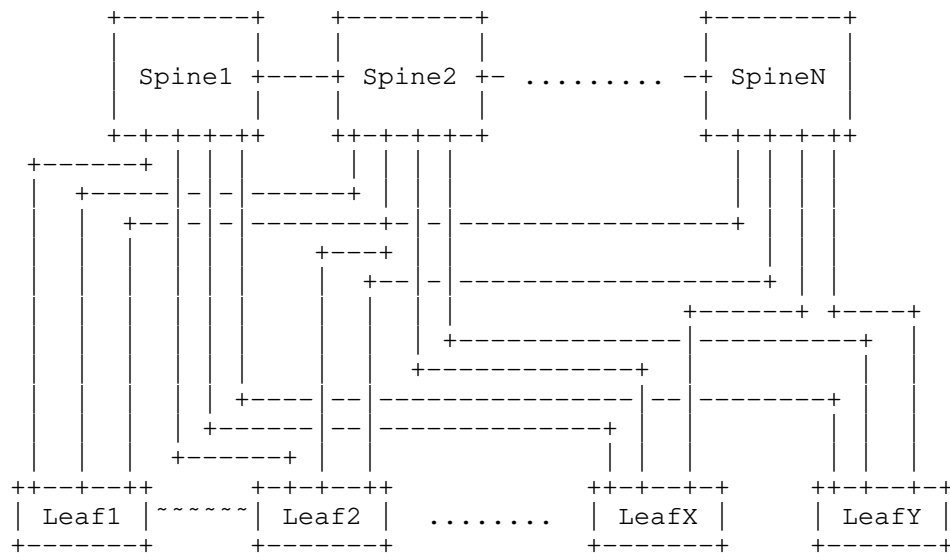


Figure 1: A Spine-Leaf Topology

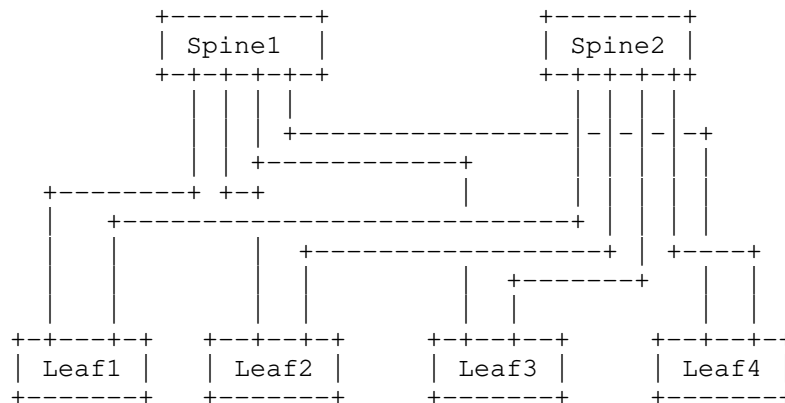


Figure 2: A CLOS Topology

### 3.2. Applicability Statement

This extension assumes the network is a Spine-Leaf topology, and it should not be applied in an arbitrary network setup. The spine nodes can be viewed as the aggregation layer of the network, and the leaf nodes as the access layer of the network. The leaf nodes use a load sharing algorithm with spine nodes as nexthops in routing and forwarding.

This extension works when the spine nodes are inter-connected, and it works with a pure CLOS or Fat Tree topology based network where the spines are NOT horizontally interconnected.

Although the example diagram in Figure 1 shows a fully meshed Spine-Leaf topology, this extension also works in the case where they are partially meshed. For instance, leaf1 through leaf10 may be fully meshed with spine1 through spine5 while leaf11 through leaf20 is fully meshed with spine4 through spine8, and all the spines are inter-connected in a redundant fashion.

This extension can also work in multi-level spine-leaf topology. The lower level spine node can be a 'leaf' node to the upper level spine node. A spine-leaf 'Tier' can be exchanged with IS-IS hello packets to allow tier X to be connected with tier X+1 using this extension. Normally tier-0 will be the TOR routers and switches if provisioned.

This extension also works with normal IS-IS routing in a topology with more than two layers of spine and leaf. For instance, in example diagrams Figure 1 and Figure 2, there can be another Core layer of routers/switches on top of the aggregation layer. From an IS-IS routing point of view, the Core nodes are not affected by this

extension and will have the complete topology and routing information just like the spine nodes. To make the network even more scalable, the Core layer can operate as a level-2 IS-IS sub-domain while the Spine and Leaf layers operate as stays at the level-1 IS-IS domain.

This extension assumes the link between the spine and leaf nodes are point-to-point, or point-to-point over LAN [RFC5309]. The links connecting among the spine nodes or the links between the leaf nodes can be any type.

### 3.3. Spine-Leaf TLVs

This extension introduces two new TLVs, the Spine-Leaf TLV and the Leaf-Set TLV. The Spine-Leaf TLV may be advertised in IS-IS Hello (IIH) PDUs; the Leaf-Set TLV may be advertised in IS-IS Circuit Scoped Link State PDUs (CS-LSP) [RFC7356]. They are used by both spine and leaf nodes in this Spine-Leaf mechanism.

#### 3.3.1. Spine-Leaf TLV

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|           Type           |      Length      |           SL Flag           |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The fields of this TLV are defined as follows:

Type: 1 octet Suggested value 151 (to be assigned by IANA)

Length: 1 octet (2 + length of sub-TLVs).

SL Flags: 16 bits

```

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-----+-----+-----+-----+-----+
| Tier |      Reserved      | T | R | L |
+-----+-----+-----+-----+

```

Tier: A value from 0 to 15. It represents the spine-leaf tier level. The value 15 is reserved to indicate the tier level is unknown. This value is only valid when the 'T' bit (see below) is set. If the 'T' bit is

clear, this value MUST be set to zero on transmission, and it MUST be ignored on receipt.

L bit (0x01): Only leaf node sets this bit. If the L bit is set in the SL flag, the node indicates it is in 'Leaf-Mode'.

R bit (0x02): Only Spine node sets this bit. If the R bit is set, the node indicates to the leaf neighbor that it can be used as the default route gateway.

T bit (0x04): If set, the value in the "Tier" field (see above) is valid.

### 3.3.2. Leaf-Set TLV

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Type           |      Length      | .. Optional Sub-TLVs
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+...

```

The Type is suggested value of 152 (to be assigned by IANA). This TLV and associated Sub-TLVs MAY appear in CS-LSP PDUs. Multiple TLVs MAY be sent.

#### 3.3.2.1. Leaf-Set Sub-TLVs

If the data center topology is a pure CLOS or Fat Tree, there are no link connections among the spine nodes. If we also assume there is not another Core layer on top of the aggregation layer, then the traffic from one leaf node to another may have a problem if there is a link outage between a spine node and a leaf node. For instance, in the diagram of Figure 2, if Leaf1 sends data traffic to Leaf3 through Spine1 node, and the Spine1-Leaf3 link is down, the data traffic will be dropped on the Spine1 node.

To address this issue spine and leaf nodes may use the sub-TLVs defined below to obtain more specific reachability information.

Two Leaf-Set sub-TLVs are defined. The Leaf-Neighbors sub-TLV and the Reachability-Req sub-TLV.

##### 3.3.2.1.1. Leaf-Neighbors Sub-TLV

This sub-TLV is used by spine nodes to advertise the current set of Leaf neighbors to Leaf nodes. The fields of this sub-TLV are defined as follows:

Type: 1 octet Suggested value 1 (to be assigned by IANA)

Length: 1 octet MUST be a multiple of 6 octets.

Leaf-Neighbors A list of IS-IS System-IDs of the leaf node neighbors of this spine node.

#### 3.3.2.1.2. Reachability-Req Sub-TLV

This sub-TLV is used by leaf nodes to request the advertisement of more specific prefix information from one or more selected spine node(s). The list of leaf nodes in this sub-TLV reflects the current set of leaf-nodes for which not all spine node neighbors have indicated the presence of connectivity in the Leaf-Neighbors sub-TLV (See Section 3.3.2.1.1). The fields of this sub-TLV are defined as follows:

Type: 1 octet Suggested value 2 (to be assigned by IANA)

Length: 1 octet. It MUST be a multiple of 6 octets.

Leaf Nodes List of IS-IS System-IDs of leaf nodes for which reachability information is being requested.

#### 3.3.3. Advertising IPv4/IPv6 Reachability

In cases where connectivity between a leaf node and a spine node is down, the leaf node MAY request reachability information from a spine node as described in Section 3.3.2.1.2. The spine node utilizes TLVs 135 [RFC5305] and TLVs 236 [RFC5308] to advertise this information. These TLVs MAY be included in CS-LSPs [RFC7356] sent from the spine to the requesting leaf node.

#### 3.3.4. Advertising Connection to RF-Leaf Node

For links between Spine and Leaf Nodes on which the Spine Node has set the R-bit and the Leaf node has set the L-bit in their respective Spine-Leaf TLVs, spine nodes MAY advertise the link with a bit in the "link-attribute" sub-TLV [RFC5029] to indicate that this link is not used for LSP flooding. This bit is named the Connect-to-RF-Leaf Node bit. This information can be used by nodes computing a flooding topology e.g., [DYNAMIC-FLOODING], to exclude the RF-Leaf nodes from the computed flooding topology.

For links between Spine and Leaf Nodes on which the Spine Node has set the R-bit and the Leaf node has set the L-bit in their respective



Spine-Leaf TLVs, leaf nodes MAY advertise the link with a bit in the "link-attribute" sub-TLV [RFC5029] to indicate that this link is to a Spine Node neighbor. This bit is named the Connect-to-RF-Spine Node bit. This information can be used by leaf nodes when deciding whether a leaf to leaf link can be used as an alternate default path when a leaf node has no connectivity to any spines. See Section 3.5.2.

### 3.4. Mechanism

Leaf nodes in a spine-leaf application using this extension are provisioned with two attributes:

- 1) Tier level of 0. This indicates the node is a Leaf Node. The value 0 is advertised in the Tier field of Spine-Leaf TLV defined above.
- 2) Flooding reduction enabled/disabled. If flooding reduction is enabled the L-bit is set to one in the Spine-Leaf TLV defined above

A spine node does not need explicit configuration. Spine nodes can dynamically discover their tier level by computing the number of hops to a leaf node. Until a spine node determines its tier level it MUST advertise level 15 (unknown tier level) in the Spine-Leaf TLV defined above. Each tier level can also be statically provisioned on the node.

When a spine node receives an IIH which includes the Spine-Leaf TLV with Tier level 0 and 'L' bit set, it labels the point-to-point interface and adjacency to be a 'Reduced Flooding Leaf-Peer (RF-Leaf)'. IIHs sent by a spine node on a link to an RF-Leaf include the Spine-Leaf TLV with the 'R' bit set in the flags field. The 'R' bit indicates to the RF-Leaf neighbor that the spine node can be used as a default routing nexthop.

There is no change to the IS-IS adjacency bring-up mechanism for Spine-Leaf peers.

A spine node blocks LSP flooding to RF-Leaf adjacencies, except for the LSP PDUs in which the IS-IS System-ID matches the System-ID of the RF-Leaf neighbor. This exception is needed since when the leaf node reboots, the spine node needs to forward to the leaf node non-purged LSPs from the RF-Leaf's previous incarnation.

Leaf nodes will perform IS-IS LSP flooding as normal to send the LSPs over all of its IS-IS adjacencies. In the case of RF-Leafs only self-originated LSPs will exist in its LSP database, and in the case

of leaf-leaf connections, there will be neighbor leaf nodes LSPs in the LSP database in addition to the self-originated LSPs.

Spine nodes will receive all the LSP PDUs in the network, including all the spine nodes and leaf nodes. It will perform Shortest Path First (SPF) as a normal IS-IS node does. There is no change to the route calculation and forwarding on the spine nodes.

The LSPs of a node only floods north bound towards the upper layer spine nodes. The default route is generated with loadsharing also towards the upper layer spine nodes.

RF-Leaf nodes do not have any LSP in the network except for its own. Therefore there is no need to perform SPF calculation on the RF-Leaf node. It only needs to download the default route with the nexthops of those Spine Neighbors which have the 'R' bit set in the Spine-Leaf TLV in IIH PDUs. IS-IS can perform equal cost or unequal cost load sharing while using the spine nodes as nexthops. The aggregated metric of the outbound interface and the 'Reverse Metric' [RFC8500] can be used for this purpose.

#### 3.4.1. Pure CLOS Topology

In a data center where the topology is pure CLOS or Fat Tree, there is no interconnection among the spine nodes, and there is not another Core layer above the aggregation layer with reachability to the leaf nodes. When flooding reduction to RF-Leafs is in use, if the link between a spine and a leaf goes down, there is then a possibility of black holing the data traffic in the network.

As in the diagram Figure 2, if the link Spine1-Leaf3 goes down, there needs to be a way for Leaf1, Leaf2 and Leaf4 to avoid the Spine1 if the destination of data traffic is to Leaf3 node.

In the above example, the Spine1 and Spine2 are provisioned to advertise the Leaf-Set sub-TLV of the Spine-Leaf TLV. Originally both Spines will advertise Leaf1 through Leaf4 as their Leaf-Set. When the Spine1-Leaf3 link is down, Spine1 will only have Leaf1, Leaf2 and Leaf4 in its Leaf-Set. This allows the other leaf nodes to know that Spine1 has lost connectivity to the leaf node of Leaf3.

Each RF-Leaf node can select another spine node to request for some prefix information associated with the lost leaf node. In this diagram of Figure 2, there are only two spine nodes (Spine-Leaf topology can have more than two spine nodes in general). Each RF-Leaf node can independently select a spine node for the leaf information. The RF-Leaf nodes will include the Info-Req sub-TLV in

the Spine-Leaf TLV in hellos sent to the selected spine node, Spine2 in this case.

The spine node, upon receiving the request from one or more leaf nodes, will find the IPv6/IPv4 prefixes advertised by the leaf nodes listed in the Info-Req sub-TLV. The spine node will use the mechanism defined in Section 3.3.2 to advertise these prefixes to the RF-Leaf node. For instance, it will include the IPv4 loopback prefix of leaf3 based on the policy configured or administrative tag attached to the prefixes. When the leaf nodes receive the more specific prefixes, they will install the advertised prefixes towards the other spine nodes (Spine2 in this example).

For instance in the data center overlay scenario, when any IP destination or MAC destination uses the leaf3's loopback as the tunnel nexthop, the overlay tunnel from leaf nodes will only select Spine2 as the gateway to reach leaf3 as long as the Spine1-Leaf3 link is still down.

In cases where multiple links or nodes fail at the same time, the RF-leaf node may need to send the Info-Req to multiple upper layer spine nodes in order to obtain reachability information for all the partially connected nodes.

This negative routing is more useful between tier 0 and tier 1 spine-leaf levels in a multi-level spine-leaf topology when the reduced flooding extension is in use. Nodes in tiers 1 or greater may have much richer topology information and alternative paths.

### 3.5. Implementation and Operation

#### 3.5.1. CSNP PDU

In Spine-Leaf extension, Complete Sequence Number PDUs (CSNP) do not need to be transmitted over the Spine-Leaf link to an RF-Leaf. Some IS-IS implementations send periodic CSNPs after the initial adjacency bring-up over a point-to-point interface. There is no need for this optimization here since the RF-Leaf does not need to receive any other LSPs from the network, and the only LSPs transmitted across the Spine-Leaf link are the leaf node LSPs.

Also in the graceful restart case[RFC5306], for the same reason, there is no need to send the CSNPs over the Spine-Leaf interface to an RF-Leaf. Spine nodes only need to set the SRMflag on the LSPs belonging to the RF-Leaf that has restarted.

### 3.5.2. Leaf to Leaf connection

Leaf to leaf node links are useful in host redundancy cases in switching networks. There are no flooding extensions required in this case. Leaf node LSPs will be exchanged over this link using the normal operation of the IS-IS Update process. In the example diagram Figure 1, Leaf1 will receive Leaf2's LSPs and Leaf2 will receive Leaf1's LSPs. Each of the Leaf nodes will in turn flood the LSPs they receive from their leaf node neighbor to their spine neighbors. Prefix reachability advertisements received from the leaf neighbor will result in the installation of more specific routes using this local Leaf-Leaf link. SPF will be performed in this case just like when the entire network only involves with those two IS-IS nodes. This does not affect the normal Spine-Leaf mechanism they perform toward the spine nodes.

Leaf to leaf connections SHOULD be limited to a single leaf neighbor.

Two modes of operation for the Leaf-Leaf link are possible and are described in the following sub-sections.

#### 3.5.2.1. Local traffic only

The leaf node sets the 'overload' bit in its LSP PDU so that spine nodes will not send traffic destined for the neighboring leaf node via its leaf node neighbor. The Leaf-Leaf link will then be used solely for local traffic between the two Leaf Nodes.

#### 3.5.2.2. Transit traffic allowed

If a leaf node becomes disconnected from all spine nodes, it is possible for spine nodes to route traffic destined for the disconnected leaf node via its leaf node neighbor. However the leaf to leaf link SHOULD be the link of last resort. To support this mode the leaf nodes do NOT set the overload bit in their LSPs and they advertise a high metric for the leaf to leaf link ( $2^{24} - 2$  is recommended). This signals to the Spine Nodes that the leaf to leaf link may be used for transit traffic, but also insures that it will not be used unless the spine node has no other path to a given leaf node.

When the leaf node is disconnected from all spine nodes it MAY install a default route towards its leaf-node neighbor in support of return traffic to the spine nodes. When doing so the leaf should validate that its leaf neighbor has at least one spine neighbor. This can be done by looking for the Connect-to-RF-Spine Node bit in the Link Attributes sub-TLVs [RFC5029] advertised in the LSPs of its leaf node neighbor.

### 3.5.3. Spine Node Hostname

This extension creates a non-reciprocal relationship between the spine node and leaf node. The spine node will receive leaf's LSP and will know the leaf's hostname, but the leaf does not have spine's LSP. This extension allows the Dynamic Hostname TLV [RFC5301] to be optionally included in spine's IIH PDU when sending to a 'Leaf-Peer'. This is useful in troubleshooting cases.

### 3.5.4. IS-IS Reverse Metric

This metric is part of the aggregated metric for leaf's default route installation with load sharing among the spine nodes. When a spine node is in 'overload' condition, it should use the IS-IS Reverse Metric TLV in IIH [RFC8500] to set this metric to maximum to discourage the leaf using it as part of the loadsharing.

In some cases, certain spine nodes may have less bandwidth in link provisioning or in real-time condition, and it can use this metric to signal to the leaf nodes dynamically.

In other cases, such as when the spine node loses a link to a particular leaf node, although it can redirect the traffic to other spine nodes to reach that destination leaf node, but it MAY want to increase this metric value if the inter-spine connection becomes overutilized, or the latency becomes an issue.

### 3.5.5. Spine-Leaf Traffic Engineering

Besides using the IS-IS Reverse Metric by the spine nodes to affect the traffic pattern for leaf default gateway towards multiple spine nodes, the IPv6/IPv4 Info-Advertise sub-TLVs can be selectively used by traffic engineering controllers to move data traffic around the data center fabric to alleviate congestion and to reduce the latency of a certain class of traffic pairs. By injecting more specific leaf node prefixes, it will allow the spine nodes to attract more traffic on some underutilized links.

### 3.5.6. Other End-to-End Services

Losing the topology information will have an impact on some of the end-to-end network services, for instance, MPLS TE or end-to-end segment routing. Some other mechanisms such as those described in PCE [RFC4655] based solution may be used. In this Spine-Leaf extension, the role of the leaf node is not too much different from the multi-level IS-IS routing while the level-1 IS-IS nodes only have the default route information towards the node which has the Attach Bit (ATT) set, and the level-2 backbone does not have any topology

information of the level-1 areas. The exact mechanism to enable certain end-to-end network services in Spine-Leaf network is outside the scope of this document.

### 3.5.7. Address Family and Topology

IPv6 Address families[RFC5308], Multi-Topology (MT)[RFC5120] and Multi-Instance (MI)[RFC8202] information is carried over the IIH PDU. Since the goal is to simplify the operation of IS-IS network, for the simplicity of this extension, the Spine-Leaf mechanism is applied the same way to all the address families, MTs and MIs.

### 3.5.8. Migration

For this extension to be deployed in existing networks, a simple migration scheme is needed. To support any leaf node in the network, all the involved spine nodes have to be upgraded first. So the first step is to migrate all the involved spine nodes to support this extension, then the leaf nodes can be enabled with 'Leaf-Mode' one by one. No flag day is needed for the extension migration.

## 4. IANA Considerations

Two new TLV codepoint is defined in this document and needs to be assigned by IANA from the "IS-IS TLV Codepoints" registry. They are referred to as the Spine-Leaf TLV and the suggested value is 151, and Leaf-Set TLV and suggested value is 152. The Spine-Leaf TLV is only to be optionally inserted in the IIH PDU, and the Leaf-Set TLV is only to be optionally inserted in Circuit Flooding Scoped LSP PDU. IANA is also requested to maintain the SL-flag bit values in the Spine-Leaf TLV, and 0x01, 0x02 and 0x04 bits are defined in this document.

Value	Name	IIH	LSP	SNP	Purge	CS-LSP
-----	-----	---	---	---	---	---
151	Spine-Leaf	y	n	n	n	n
152	Leaf-Set	n	n	n	n	y

This document also proposes to have the Dynamic Hostname TLV, already assigned as code 137, to be allowed in IIH PDU.

Value	Name	IIH	LSP	SNP	Purge
-----	-----	---	---	---	---
137	Dynamic Name	y	y	n	y

This documents requests IANA to create a new registry under the IS-IS TLV Codepoints registry. The suggested name of the registry is "Sub-

TLVs for TLV 152 (Leaf-Set TLV)". Initial contents of the new registry is defined below:

Value	Name
-----	-----
0	Reserved
1	Leaf Neighbors
2	Reachability Req
3-255	Unassigned

This document also requests that IANA allocate from the registry of link-attribute two new bit values for sub-TLV 19 of TLV 22 (Extended IS reachability TLV).

Value	Name	Reference
-----	-----	-----
0x4	Connect to RF-Leaf Node	This document
0x8	Connect to RF-Spine Node	This document

## 5. Security Considerations

Security concerns for IS-IS are addressed in [ISO10589], [RFC5304], [RFC5310], and [RFC7602]. This extension does not raise additional security issues.

## 6. Acknowledgments

The authors would like to thank Tony Przygienda and Lukas Krattiger for their discussion and contributions. The authors also would like to thank Acee Lindem, Russ White, Christian Hopps and Aijun Wang for their review and comments of this document.

## 7. References

### 7.1. Normative References

- [ISO10589] ISO "International Organization for Standardization", "Intermediate system to Intermediate system intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473), ISO/IEC 10589:2002, Second Edition.", Nov 2002.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

- [RFC5029] Vasseur, JP. and S. Previdi, "Definition of an IS-IS Link Attribute Sub-TLV", RFC 5029, DOI 10.17487/RFC5029, September 2007, <<https://www.rfc-editor.org/info/rfc5029>>.
- [RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)", RFC 5120, DOI 10.17487/RFC5120, February 2008, <<https://www.rfc-editor.org/info/rfc5120>>.
- [RFC5301] McPherson, D. and N. Shen, "Dynamic Hostname Exchange Mechanism for IS-IS", RFC 5301, DOI 10.17487/RFC5301, October 2008, <<https://www.rfc-editor.org/info/rfc5301>>.
- [RFC5304] Li, T. and R. Atkinson, "IS-IS Cryptographic Authentication", RFC 5304, DOI 10.17487/RFC5304, October 2008, <<https://www.rfc-editor.org/info/rfc5304>>.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <<https://www.rfc-editor.org/info/rfc5305>>.
- [RFC5306] Shand, M. and L. Ginsberg, "Restart Signaling for IS-IS", RFC 5306, DOI 10.17487/RFC5306, October 2008, <<https://www.rfc-editor.org/info/rfc5306>>.
- [RFC5308] Hopps, C., "Routing IPv6 with IS-IS", RFC 5308, DOI 10.17487/RFC5308, October 2008, <<https://www.rfc-editor.org/info/rfc5308>>.
- [RFC5310] Bhatia, M., Manral, V., Li, T., Atkinson, R., White, R., and M. Fanto, "IS-IS Generic Cryptographic Authentication", RFC 5310, DOI 10.17487/RFC5310, February 2009, <<https://www.rfc-editor.org/info/rfc5310>>.
- [RFC7356] Ginsberg, L., Previdi, S., and Y. Yang, "IS-IS Flooding Scope Link State PDUs (LSPs)", RFC 7356, DOI 10.17487/RFC7356, September 2014, <<https://www.rfc-editor.org/info/rfc7356>>.
- [RFC7602] Chunduri, U., Lu, W., Tian, A., and N. Shen, "IS-IS Extended Sequence Number TLV", RFC 7602, DOI 10.17487/RFC7602, July 2015, <<https://www.rfc-editor.org/info/rfc7602>>.
- [RFC8202] Ginsberg, L., Previdi, S., and W. Henderickx, "IS-IS Multi-Instance", RFC 8202, DOI 10.17487/RFC8202, June 2017, <<https://www.rfc-editor.org/info/rfc8202>>.



- [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>>.

## 7.2. Informative References

- [DYNAMIC-FLOODING]  
Li, T., "Dynamic Flooding on Dense Graphs", draft-li-dynamic-flooding (work in progress), 2018.
- [RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <<https://www.rfc-editor.org/info/rfc4655>>.
- [RFC5309] Shen, N., Ed. and A. Zinin, Ed., "Point-to-Point Operation over LAN in Link State Routing Protocols", RFC 5309, DOI 10.17487/RFC5309, October 2008, <<https://www.rfc-editor.org/info/rfc5309>>.

## Authors' Addresses

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

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

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

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

Sanjay Thyamagundalu

Email: [tsanjay@gmail.com](mailto:tsanjay@gmail.com)

Link State Routing  
Internet-Draft  
Intended status: Standards Track  
Expires: May 4, 2020

K. Talaulikar  
P. Psenak  
Cisco Systems, Inc.  
A. Fu  
Bloomberg  
M. Rajesh  
Juniper Networks  
November 1, 2019

OSPF Strict-Mode for BFD  
draft-ketant-lsr-ospf-bfd-strict-mode-03

Abstract

This document specifies the extensions to OSPF that enables a router and its neighbor to signal their intention to use Bidirectional Forwarding Detection (BFD) for their adjacency using link-local advertisement between them. The signaling of this BFD enablement, allows the router to block and not allow the establishment of adjacency with its neighbor router until a BFD session is successfully established between them. The document describes this OSPF "strict-mode" of BFD establishment as a prerequisite to adjacency formation.

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 May 4, 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. LLS B-bit Flag . . . . .	3
3. Local Interface IPv4 Address TLV . . . . .	4
4. Procedures . . . . .	4
4.1. OSPFv3 IPv4 Address-Family Specifics . . . . .	6
4.2. Graceful Restart Considerations . . . . .	6
5. Operations & Management Considerations . . . . .	6
6. Backward Compatibility . . . . .	7
7. IANA Considerations . . . . .	7
8. Security Considerations . . . . .	8
9. Acknowledgements . . . . .	8
10. References . . . . .	8
10.1. Normative References . . . . .	8
10.2. Informative References . . . . .	9
Authors' Addresses . . . . .	9

#### 1. Introduction

Bidirectional Forwarding Detection (BFD) [RFC5880] enables routers to monitor dataplane connectivity over links between them and to detect faults in the bidirectional path between them. This capability is leveraged by routing protocols like Open Shortest Path First (OSPFv2) [RFC2328] and OSPFv3 [RFC5340] to detect connectivity failures for their adjacencies and trigger the rerouting of traffic around this failure more quickly than their periodic hello messaging based detection mechanism.

The use of BFD for monitoring routing protocols adjacencies is described in [RFC5882]. When BFD monitoring is enabled for OSPF

adjacencies, the BFD session is bootstrapped based on the neighbor address information discovered by the exchange of OSPF hello messages. Faults in the bidirectional forwarding detected via BFD then result in the bringing down of the OSPF adjacency. Note that it is possible in some failure scenarios for the network to be in a state such that the OSPF adjacency is capable of coming up, but the BFD session cannot be established, and, more particularly, data cannot be forwarded. In certain other scenarios, a degraded or poor quality link may result in OSPF adjacency formation to succeed only to result in BFD session establishment not being successful or the BFD session going down frequently due to its faster detection mechanism.

To avoid such situations which result in routing churn in the network, it would be beneficial not to allow OSPF to establish a neighbor adjacency until the BFD session is successfully established and stabilized. However, this would preclude the OSPF operation in an environment in which not all OSPF routers support BFD and are enabled for BFD monitoring. A solution would be to block the establishment of OSPF adjacencies if both systems are willing to establish a BFD session but a BFD session cannot be established. Such a mode of BFD use by OSPF is referred to as "strict-mode" wherein BFD session establishment becomes a prerequisite for OSPF adjacency coming up.

This document specifies the OSPF protocol extensions using link-local signaling (LLS) [RFC5613] for a router to indicate to its neighbor the willingness to establish a BFD session in the "strict-mode". It also introduces an extension for OSPFv3 link-local signaling of interface IPv4 address when used for IPv4 address-family (AF) instance to enable discovery of the IPv4 addresses for BFD session setup.

A similar functionality for IS-IS is specified [RFC6213].

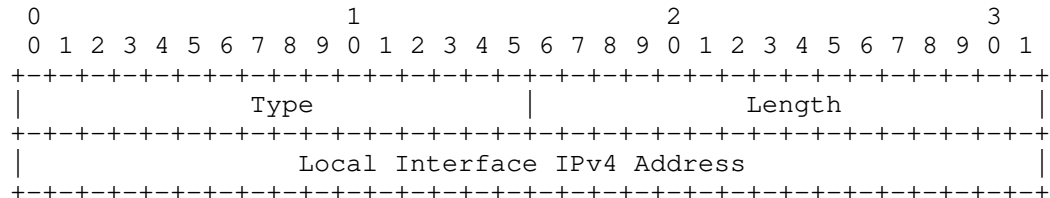
## 2. LLS B-bit Flag

A new B-bit is defined in the LLS Type 1 Extended Options and Flags field. This bit is defined for the LLS block included in Hello packets and indicates that BFD is enabled on the link and that the router supports BFD strict-mode. Section 7 describes the position of this new B-bit.

A router MUST include the LLS block with the LLS Type 1 Extended Options and Flags TLV with the B-bit set its Hello messages when BFD is enabled on the link.

### 3. Local Interface IPv4 Address TLV

The Local Interface IPv4 Address TLV is a new LLS TLV meant for OSPFv3 protocol operations for IPv4 AF instances [RFC5838]. It has following format:



where:

Type: TBD, suggested value 21

Length: 4 octet

Local Interface IPv4 Address: The primary IPv4 address of the local interface.

### 4. Procedures

A router supporting BFD strict-mode advertises this capability through its hello messages as described in Section 2 above. When a router supporting BFD strict-mode, detects a new neighbor router that also supports BFD strict-mode, then it proceeds to establish adjacency with that neighbor as described further in this section.

This document updates the OSPF neighbor state machine as described in [RFC2328] specifically the operations related to the Init state as below when BFD strict-mode is used:

Init (without BFD strict-mode)

In this state, an Hello packet has recently been seen from the neighbor. However, bidirectional communication has not yet been established with the neighbor (i.e., the router itself did not appear in the neighbor's Hello packet). All neighbors in this state (or higher) are listed in the Hello packets sent from the associated interface.

Init (with BFD strict-mode)

In this state, an Hello packet has recently been seen from the neighbor. However, bidirectional communication has not yet been

established with the neighbor (i.e., the router itself did not appear in the neighbor's Hello packet). A BFD session establishment to the neighbor is requested, if not already done (e.g. in the event of transition from 2-way state). All neighbors in higher than Init state and those in Init state with BFD session up are listed in the Hello packets sent from the associated interface.

Whenever the neighbor state transitions to Down state, the removal of the BFD session associated with that neighbor SHOULD be requested by OSPF and the session re-setup SHOULD similarly be requested by OSPF after transitioning into Init state. This may result in the deletion and creation of BFD session respectively when OSPF is the only client interested in BFD session to the neighbor address.

An implementation MUST NOT wait for BFD session establishment in Init state unless BFD strict-mode is enabled on the router and the specific neighbor indicates BFD strict-mode capability via its Hello messages. When BFD is enabled, but the strict-mode of operation cannot be used, then an implementation SHOULD start the BFD session establishment only in 2-Way or higher state. This makes it possible for router to operate a mix of BFD operation in strict-mode or normal mode across different interfaces or even different neighbors on the same multi-access LAN interface.

Once the OSPF state machine has moved beyond the Init state, any change in the B-bit advertised in subsequent Hello messages MUST NOT result in any trigger in either the OSPF adjacency or the BFD session management (i.e. the B-bit is considered only when in the Init state). The disabling of BFD (or BFD strict-mode) on a router would result in its not setting the B-bit in its subsequent Hello messages. The disabling of BFD strict-mode has no change on the BFD operations and would not result in bringing down of any established BFD session. The disabling of BFD would result in the BFD session brought down due to Admin reason and hence would not bring down the OSPF adjacency.

When BFD is enabled on an interface over which we already have an existing OSPF adjacency, it would result in the router setting the B-bit in its subsequent Hello messages. If the adjacency is already up (i.e. in its terminal state of Full or 2-way with non-DR routers on a LAN) with a neighbor that also support BFD strict-mode, then an implementation SHOULD NOT bring this adjacency down and instead use the BFD strict-mode of operations after the next transition into Init state. However, if the adjacency is not up, then an implementation MAY bring such an adjacency down so it can use the BFD strict-mode for its bring up.

#### 4.1. OSPFv3 IPv4 Address-Family Specifics

The multiple AF support in OSPFv3 [RFC5838] requires the use of IPv6 link-local address as source address for hello packets even when forming adjacencies for IPv4 AF instances. In most deployments of OSPFv3 IPv4 AF, it is required that BFD be used to monitor and verify the IPv4 data plane connectivity between the routers on the link and hence the BFD session is setup using IPv4 neighbor addresses. The IPv4 neighbor address on the interface is learnt only later in the adjacency formation phase when the neighbor's Link-LSA is received. This results in the setup of the BFD session either after the adjacency is established or much later in the adjacency formation sequence.

To enable the BFD operations in strict-mode, it is necessary for a router to learn its neighbor's IPv4 link address during the Init state of adjacency formation (ideally when it receives the first hello). The use of the Local Interface IPv4 Address TLV (as defined in Section 3) in the LLS block of the OSPFv3 Hello messages for IPv4 AF instances makes this possible. Implementations that support strict-mode of BFD operations for OSPFv3 IPv4 AF instances MUST include the Local Interface IPv4 Address TLV in the LLS block of their hello messages whenever the B-bit is set. A receiver MUST ignore the B-bit (i.e. not operate in BFD strict mode) unless the Local Interface IPv4 Address TLV is present in OSPFv3 Hello message for IPv4 AF instances.

#### 4.2. Graceful Restart Considerations

An implementation needs to handle scenarios where both graceful restart (GR) and the strict-mode of BFD operations are deployed together. The GR aspects discussed in [RFC5882] also apply with strict-mode of operations. In addition to that, since the OSPF adjacency formation is held up until the BFD session establishment in the strict-mode of operation, the resultant delay in adjacency formation may affect or break the GR based recovery. In such cases, it is RECOMMENDED that the GR timers are setup such that they provide sufficient time to cover for normal BFD session establishment delays.

#### 5. Operations & Management Considerations

An implementation SHOULD report the BFD session status along with the OSPF Init adjacency state when operating in BFD strict-mode and perform logging operations on state transitions to include the BFD events. This allows an operator to detect scenarios where an OSPF adjacency may be stuck waiting for BFD session establishment.

In network deployments with noisy links or those with packet loss, BFD sessions may flap frequently. In such scenarios, OSPF strict-mode for BFD may be deployed in conjunction with an BFD dampening or hold-down mechanism to help avoid frequent adjacency flaps due BFD causing routing churn.

## 6. Backward Compatibility

An implementation MUST support OSPF adjacency formation and operations with a neighbor router that does not advertise the BFD strict-mode capability - both when that neighbor router does not support BFD and when it does support BFD but not in the strict-mode of operation as described in this document. Implementations MAY provide an option to specifically enable BFD operations only in the strict-mode in which case, OSPF adjacency with a neighbor that does not support BFD strict-mode would not be established successfully. Implementations MAY provide an option to disable BFD strict-mode which results in the router not advertising the B-bit and BFD operations being performed in the same way as before this specification.

The signaling specified in this document happens at a link-local level between routers on that link. A router which does not support this specification would ignore the B-bit in the LLS block of hello messages from its neighbors and continue to bootstrap BFD sessions, if enabled, without holding back the OSPF adjacency formation. Since the router which does not support this specification would not have set the B-bit in the LLS block of its own hello messages, its neighbor routers that support this specification would not use BFD strict-mode with it. As a result, the behavior would be the same as before this specification. Therefore, there are no backward compatibility related issues or considerations that need to be taken care of when implementing this specification.

## 7. IANA Considerations

This specification updates Link Local Signaling TLV Identifiers registry.

Following values are requested for allocation:

- o B-bit from "LLS Type 1 Extended Options and Flags" registry at bit position 0x00000010.
- o TBD (Suggested value 21) - Local Interface IPv4 Address TLV



## 8. Security Considerations

The security considerations for "OSPF Link-Local Signaling" [RFC5613] also apply to the extension described in this document. Inappropriate use of the B-bit in the LLS block of an OSPF hello message could prevent an OSPF adjacency from forming or lead to failure to detect bidirectional forwarding failures. If authentication is being used in the OSPF routing domain [RFC5709][RFC7474], then the Cryptographic Authentication TLV [RFC5613] SHOULD also be used to protect the contents of the LLS block.

## 9. Acknowledgements

The authors would like to acknowledge the review and inputs from Acee Lindem, Manish Gupta, Balaji Ganesh and Rajesh M.

The authors would like to acknowledge Dylan van Oudheusden for highlighting the problems in using strict-mode for BFD session for IPv4 AF instance with OSPFv3 and Baalajee S for his suggestions on the approach to address it.

## 10. References

### 10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <<https://www.rfc-editor.org/info/rfc2328>>.
- [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>>.
- [RFC5613] Zinin, A., Roy, A., Nguyen, L., Friedman, B., and D. Yeung, "OSPF Link-Local Signaling", RFC 5613, DOI 10.17487/RFC5613, August 2009, <<https://www.rfc-editor.org/info/rfc5613>>.
- [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>>.

- [RFC5882] Katz, D. and D. Ward, "Generic Application of Bidirectional Forwarding Detection (BFD)", RFC 5882, DOI 10.17487/RFC5882, June 2010, <<https://www.rfc-editor.org/info/rfc5882>>.
- [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>>.

## 10.2. Informative References

- [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>>.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <<https://www.rfc-editor.org/info/rfc5880>>.
- [RFC6213] Hopps, C. and L. Ginsberg, "IS-IS BFD-Enabled TLV", RFC 6213, DOI 10.17487/RFC6213, April 2011, <<https://www.rfc-editor.org/info/rfc6213>>.
- [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>>.

## Authors' Addresses

Ketan Talaulikar  
Cisco Systems, Inc.  
India

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

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

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

Albert Fu  
Bloomberg  
USA

Email: [afu14@bloomberg.net](mailto:afu14@bloomberg.net)

Rajesh M  
Juniper Networks  
India

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

Link State Routing  
Internet-Draft  
Intended status: Standards Track  
Expires: February 21, 2020

K. Talaulikar  
P. Psenak  
Cisco Systems, Inc.  
H. Johnston  
AT&T Labs  
August 20, 2019

OSPF Reverse Metric  
draft-ketant-lsr-ospf-reverse-metric-02

Abstract

This document specifies the extensions to OSPF that enables a router to signal to its neighbor the metric that the neighbor should use towards itself using link-local advertisement between them. The signalling of this reverse metric, to be used on link(s) towards itself, allows a router to influence the amount of traffic flowing towards itself and in certain use-cases enables routers to maintain symmetric metric on both sides of a link between them.

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 February 21, 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. Use Cases . . . . .	3
2.1. Symmetrical Metric Based on Reference Bandwidth . . . . .	3
2.2. Adaptive Metric Signaling . . . . .	4
3. Solution . . . . .	5
4. LLS Reverse Metric TLV . . . . .	6
5. LLS Reverse TE Metric TLV . . . . .	6
6. Procedures . . . . .	7
7. Backward Compatibility . . . . .	9
8. IANA Considerations . . . . .	9
9. Security Considerations . . . . .	9
10. Contributors . . . . .	10
11. Acknowledgements . . . . .	10
12. References . . . . .	10
12.1. Normative References . . . . .	10
12.2. Informative References . . . . .	11
Authors' Addresses . . . . .	11

## 1. Introduction

Routers running the Open Shortest Path First (OSPFv2) [RFC2328] and OSPFv3 [RFC5340] routing protocols originate a Router-LSA (Link State Advertisement) that describes all its links to its neighbors and includes a metric which indicates its "cost" of reaching the neighbor over that link. Consider two routers R1 and R2 that are connected via a link. The metric for this link in direction R1->R2 is configured on R1 and in the direction R2->R1 is configured on R2. Thus the configuration on R1 influences the traffic that it forwards towards R2 but does not influence the traffic that it may receive from R2 on that same link.

This document describes certain use-cases where it is desirable for a router to be able to signal what we call as the "reverse metric" (RM) to its neighbor to adjust the routing metric on the inbound direction. When R1 signals its reverse metric on its link to R2, then R2 advertises this value as its metric to R1 in its Router-LSA instead of its locally configured value. Once this information is part of the topology then all other routers do their computation using this value which results in the desired change in traffic distribution that R1 wanted to achieve towards itself over the link from R2.

This document proposes an extension to OSPF link-local signaling (LLS) [RFC5613] for signalling the OSPF reverse metric using the LLS Reverse Metric TLV in Section 4, the reverse Traffic Engineering (TE) metric [RFC3630] using the LLS Reverse TE Metric TLV in Section 5 and describes the related procedures in section Section 6.

## 2. Use Cases

This section describes certain use-cases that OSPF reverse metric helps to address. The usage of OSPF reverse metric need not be limited to these cases and is intended to be a generic mechanism.

### 2.1. Symmetrical Metric Based on Reference Bandwidth

Certain OSPF implementations and deployments deduce the metric of links based on their bandwidth using a reference bandwidth. The OSPF MIB [RFC4750] has `ospfReferenceBandwidth` that is used by entries in the `ospfIfMetricTable`. This mechanism is leveraged in deployments where the link metrics get lowered or increased as bandwidth capacity is removed or added e.g. consider layer-2 links bundled as a layer-3 interface on which OSPF is enabled. In the situations where these layer-2 links are directly connected to the two routers, the link and bandwidth availability is detected and updated on both sides. This allows for schemes where the metric is maintained to be symmetric in both directions based on the bandwidth.

Now consider variation of the same deployment where the links between routers are not directly connected and instead are provisioned over a layer-2 network consisting of switches or other mechanisms for a layer-2 emulation. In such scenarios, as show in Figure 1, the router on one side of the link would not detect when the neighboring router has lost one of its layer-2 link and has reduced capacity to its layer-2 switch. Note that the number of links and their capacities on the router R0 may not be the same as those on R1, R2 and R3. The left hand side diagram shows the actual physical topology in terms of the layer-2 links while the right hand side diagram shows the logical layer-3 link topology between the routers.

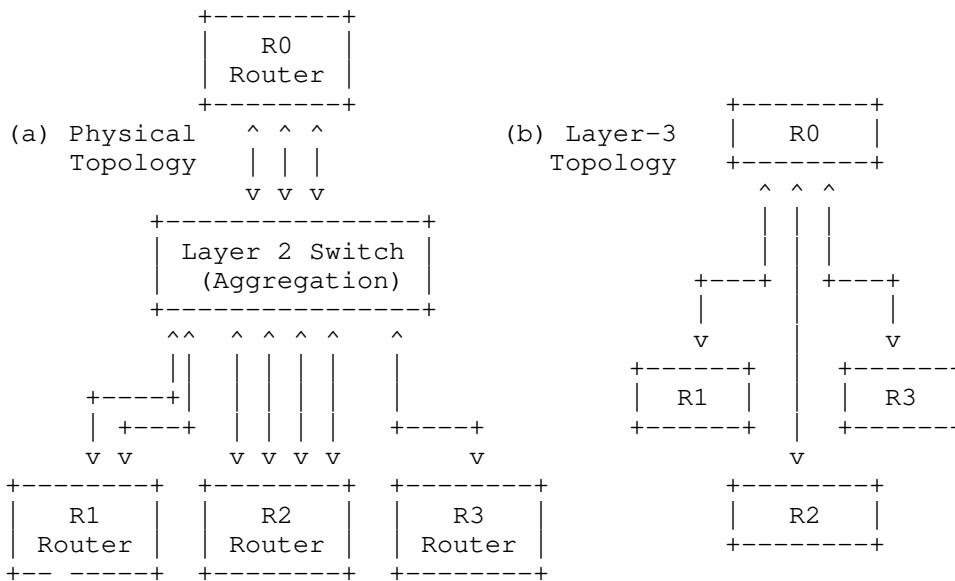


Figure 1: Routers Interconnected over Layer-2 Network

In such a scenario, the amount of traffic that can be forwarded in bidirectional manner between say R0 and R1 is dictated by the lower of the link capacity of R0 and R1 to the layer-2 transport network. In this scenario, when one of the link from R1 to the switch goes down, it would increase its link metric to R0 from say 20 to 40. However, similarly R0 also needs to increase its link metric to R1 as well from 20 to 40 as otherwise, the traffic will hit congestion and get dropped.

When R1 has the ability to signal the OSPF reverse metric of 40 towards itself to R0, then R0 can also update its metric without any manual intervention to ensure the correct traffic distribution. Consider if some destinations were reachable from R0 via R1 previously and this automatic metric adjustment now makes some of those destinations reachable from R0 via R3. This allows some traffic load on the link R0 to R1 to now flow via R3 to these destinations.

## 2.2. Adaptive Metric Signaling

Now consider another deployment scenario where, as show in Figure 2, two routers AGGR1 and AGGR2 are connected to a bunch of routers R1 thru RN that are dual homed to them and aggregating the traffic from them towards a core network. At some point T, AGGR1 loses some of its capacity towards the core or is facing some congestion issue

towards the core and it needs to reduce the traffic going through it and perhaps redirect some of that load via AGGR2 which is not facing a similar issue. Altering its own metric towards Rx routers would influence the traffic flowing through it in the direction from core to the Rx but not the other way around as desired.

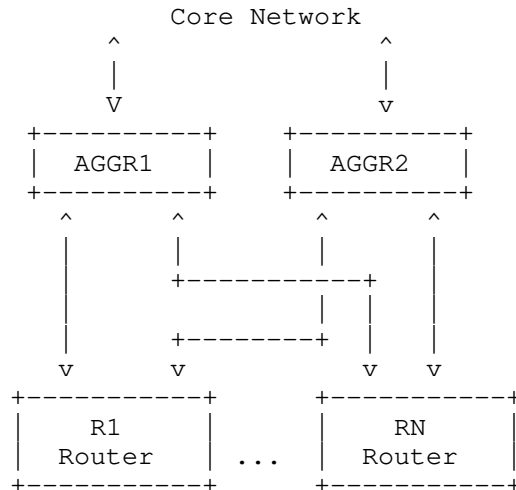


Figure 2: Adaptive Metric for Dual Gateways

In such a scenario, the AGGR1 router could signal an incremental value of OSPF reverse metric towards some or all of the Rx routers. When the Rx routers apply this signaled reverse metric offset value to the original metric on their links towards AGGR1 then the path via AGGR2 becomes a better path causing traffic towards the core getting diverted away from it. Note that the reverse metric mechanism allows such adaptive metric changes to be applied on the AGGR1 as opposed to being provisioning statically on the possibly large number of Rx routers.

### 3. Solution

To address the use-cases described earlier and to allow an OSPF router to indicate its reverse metric for a specific point-to-point or point-to-multipoint link to its neighbor, this document proposes to extend OSPF link-local signaling to advertise the Reverse Metric TLV in OSPF Hello packets. This ensures that the RM signaling is scoped ONLY to each specific link individually. The router continues to include the Reverse Metric TLV in its Hello packets on the link as long as it needs its neighbor to use that metric value towards itself. Further details of the procedures involve are specified in Section 6.



The RM signaling specified in this document is not required for broadcast or non-broadcast-multi-access (NBMA) links since the same objective is achieved there using the OSPF Two-Part Metric mechanism [RFC8042].

#### 4. LLS Reverse Metric TLV

The Reverse Metric TLV is a new LLS TLV. It has following format:

```

      0               1               2               3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     |                                     |
|               Type                 |               Length                 |
+-----+-----+-----+-----+-----+-----+-----+-----+
|      MTID      |  Flags  | O | H |      Reverse Metric      |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

where:

Type: TBD, suggested value 19

Length: 4 octet

MTID : the multi-topology identifier of the link ([RFC4915])

Flags: 1 octet, following are defined currently and the rest MUST be set to 0 and ignored on reception.

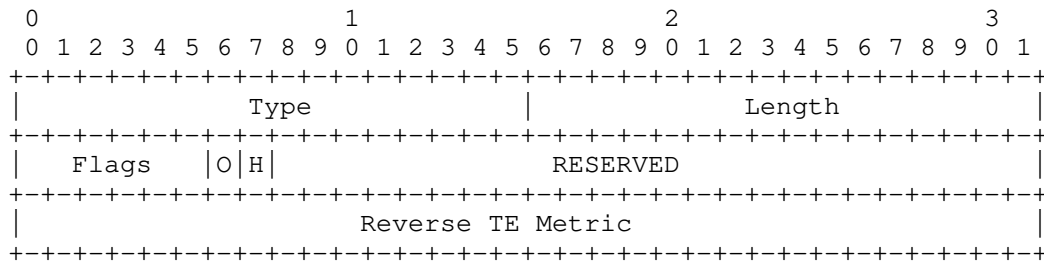
\* H (0x1) : Indicates that neighbor should use value only if higher than its current metric value in use

\* O (0x2) : Indicates that the reverse metric value provided is an offset that is to be added to the original metric

Reverse Metric: 2 octets, the value or offset of reverse metric to be used

#### 5. LLS Reverse TE Metric TLV

The Reverse TE Metric TLV is a new LLS TLV. It has following format:



where:

Type: TBD, suggested value 20

Length: 4 octet

Flags: 1 octet, following are defined currently and the rest MUST be set to 0 and ignored on reception.

- \* H (0x1) : Indicates that neighbor should use value only if higher than its current TE metric value in use
- \* O (0x2) : Indicates that the reverse TE metric value provided is an offset that is to be added to the original TE metric

RESERVED: 24-bit field. SHOULD be set to 0 on transmission and MUST be ignored on receipt.

Reverse TE Metric: 4 octets, the value or offset of reverse traffic engineering metric to be used

## 6. Procedures

When a router needs to signal a RM value that its neighbor(s) should use towards itself, it includes the Reverse Metric TLV in the LLS block of its hello messages sent on the link and continues to include this TLV for as long as it needs it's neighbor to use this value. The mechanisms used to determine the value to be used for the RM is specific to the implementation and use-case and is outside the scope of this document. e.g. in the use-case related to symmetric metric described in Section 2.1, the RM value may be derived based on the router's link's bandwidth with respect to the reference bandwidth.

A router receiving a hello packet from its neighbor that contains the Reverse Metric TLV on its link SHOULD use the RM value to derive the metric for the link in its Router-LSA to the advertising router.

When the O flag is set, the value in the TLV needs to be added to the existing original metric provisioned on the link to derive the new metric value to be used. When the O flag is clear, the value in the TLV should be directly used as the metric to be used. When H flag is set and O flag is clear, this is done only when the RM value signaled is higher than the provisioned metric value being used already. This mechanism applies only for point-to-point, point-to-multipoint and hybrid broadcast point-to-multipoint ([RFC6845]) links. For broadcast and NBMA links the OSPF Two-Part Metric mechanism [RFC8042] should be used in similar use-cases.

Implementations SHOULD provide a configuration option to enable the signaling of RM from a router to its neighbors and MAY provide a configuration option to disable the acceptance of the RM from its neighbors.

A router stops including the Reverse Metric TLV in its hello messages when it needs its neighbors to go back to using their own provisioned metric values. When that happens, a router which had modified its metric in response to receiving a Reverse Metric TLV from its neighbor should revert back to using its original provisioned metric value.

In certain scenarios, it is possible that two or more routers start the RM signaling on the same link. This could create collision scenarios. The following rules MUST be adopted by routers to ensure that there is no instability in the network due to churn in their metric due to signaling of RM:

- o The RM value that is signaled by a router to its neighbor MUST NOT be derived from the reverse metric being signaled by any of its neighbor on any of its links.
- o The RM value that is signaled by a router MUST NOT be derived from its own metric which has been modified on account of a RM signaled from any of its neighbors on any of its links. RM signaling from other routers can affect the router's own metric advertised in its Router-LSA. When deriving the RM values that a router signals to its neighbors, it should use its "original" local metric values not influenced by any RM signaling.

Based on these rules, a router MUST never start or stop or change its RM metric signaling based on the RM metric signaling initiated by some other router. Based on the local configuration policy, each router would end up accepting the RM value signaled by its neighbor and there would be no churn of metrics on the link or the network on account of RM signaling.

In certain use-case as described in Section 2.1 when symmetrical metrics are desired, the RM signaling can be enabled on routers on either ends of a link. In other use-cases as described in Section 2.2 RM signaling may need to be enabled on only router at one end of a link.

When using multi-topology routing with OSPF [RFC4915] a router MAY include multiple instances of the Reverse Metric TLV in the LLS block of its hello message - one for each of the topology for which it desires to signal the reserve metric for.

In certain scenarios, the OSPF router may also require the modification of the TE metric being advertised by its neighbor router towards itself in the inbound direction. The Reverse TE Metric TLV, using similar procedures as described above, MAY be used to signal the reverse TE metric by a router. The neighbor SHOULD use the reverse TE metric value to derive the TE metric to be used in the TE Metric sub-TLV of the Link TLV in its TE Opaque LSA [RFC3630].

## 7. Backward Compatibility

The signaling specified in this document happens at a link-local level between routers on that link. A router which does not support this specification would ignore the Reverse Metric and Reverse TE Metric LLS TLVs and take no actions to updates its metric in the other LSAs. As a result, the behavior would be the same as before this specification. Therefore, there are no backward compatibility related issues or considerations that need to be taken care of when implementing this specification.

## 8. IANA Considerations

This specification updates Link Local Signalling TLV Identifiers registry.

Following values are requested for allocation:

- o TBD (Suggested value 19) - Reverse Metric TLV
- o TBD (Suggested value 20) - Reverse TE Metric TLV

## 9. Security Considerations

The security considerations for "OSPF Link-Local Signaling" [RFC5613] also apply to the extension described in this document. The usage of the reverse metric TLVs is to alter the metrics used by routers on the link and influence the flow and routing of traffic over the network. Hence, modification of the Reverse Metric and Reverse TE

Metric TLVs may result in misrouting of traffic. If authentication is being used in the OSPF routing domain [RFC5709][RFC7474], then the Cryptographic Authentication TLV [RFC5613] SHOULD also be used to protect the contents of the LLS block.

Receiving a malformed LLS Reverse Metric or Reverse TE Metric TLVs MUST NOT result in a hard router or OSPF process failure. The reception of malformed LLS TLVs or sub-TLVs SHOULD be logged, but such logging MUST be rate-limited to prevent denial-of-service (DoS) attacks.

## 10. Contributors

Thanks to Jay Karthik for his contributions on the use-cases related to symmetric metric and the review of the solution.

## 11. Acknowledgements

## 12. References

### 12.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <<https://www.rfc-editor.org/info/rfc2328>>.
- [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>>.
- [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>>.
- [RFC5613] Zinin, A., Roy, A., Nguyen, L., Friedman, B., and D. Yeung, "OSPF Link-Local Signaling", RFC 5613, DOI 10.17487/RFC5613, August 2009, <<https://www.rfc-editor.org/info/rfc5613>>.
- [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>>.

## 12.2. Informative References

- [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>>.
- [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>>.
- [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>>.
- [RFC6845] Sheth, N., Wang, L., and J. Zhang, "OSPF Hybrid Broadcast and Point-to-Multipoint Interface Type", RFC 6845, DOI 10.17487/RFC6845, January 2013, <<https://www.rfc-editor.org/info/rfc6845>>.
- [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>>.
- [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>>.

## Authors' Addresses

Ketan Talaulikar  
Cisco Systems, Inc.  
India

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

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

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

Hugh Johnston  
AT&T Labs  
USA

Email: [hugh\\_johnston@labs.att.com](mailto:hugh_johnston@labs.att.com)

Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: December 30, 2018

T. Li  
Arista Networks  
June 28, 2018

Hierarchical IS-IS  
draft-li-hierarchical-isis-00

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 December 30, 2018.

Copyright Notice

Copyright (c) 2018 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 . . . . .	3
2.2. PDU Type . . . . .	4
3. Additional PDUs . . . . .	4
3.1. LAN IS to IS hello PDU (LAN-HELLO-PDU) . . . . .	4
3.2. Point-to-point IS to IS hello PDU (P2P-HELLO-PDU) . . . . .	4
3.3. Level n Link State PDU (Ln-LSP-PDU) . . . . .	4
3.4. Level n complete sequence numbers PDU (Ln-CSNP-PDU) . . . . .	5
3.5. Level n partial sequence numbers PDU (Ln-PSNP-PDU) . . . . .	5
4. Inheritance of TLVs . . . . .	5
5. Acknowledgements . . . . .	6
6. IANA Considerations . . . . .	6
6.1. PDU Type . . . . .	6
6.2. New PDUs . . . . .	6
7. Security Considerations . . . . .	7
8. Normative References . . . . .	7
Author's Address . . . . .	7

## 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 by creating additional hierarchy. 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 some sense to expand the design for the future.

The modifications described herein are designed to be fully backward compatible.

Section references in this document are references to sections of IS-IS [ISO10589].

### 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. 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. LAN IS to IS hello PDU (LAN-HELLO-PDU)

The 'LAN IS to IS hello PDU' (LAN-HELLO-PDU) is identical in format to the 'Level 2 LAN IS to IS hello PDU' (Section 9.6), except that the PDU Type has value AAA. The LAN-HELLO-PDU MUST be used instead of the 'Level 1 LAN IS to IS hello PDU' (Section 9.5) or the 'Level 2 LAN IS to IS hello PDU' on any circuit that has one or more of Level 3 through Level 8 enabled.

### 3.2. Point-to-point IS to IS hello PDU (P2P-HELLO-PDU)

The 'Point-to-point IS to IS hello PDU' can be used on circuits of any Level without modification.

### 3.3. Level n Link State PDU (Ln-LSP-PDU)

The 'Level n Link State PDU' (Ln-LSP-PDU) has the same format as the 'Level 2 Link State PDU' (Section 9.9), except for the PDU Type. The PDU Types for Levels 3 through 8 are defined as follows:

Level 3 (L3-LSP-PDU): BBB

Level 4 (L4-LSP-PDU): CCC

Level 5 (L5-LSP-PDU): DDD

Level 6 (L6-LSP-PDU): EEE

Level 7 (L7-LSP-PDU): FFF

Level 8 (L8-LSP-PDU): GGG

### 3.4. Level n complete sequence numbers PDU (Ln-CSNP-PDU)

The 'Level n complete sequence numbers PDU' (Ln-CSNP-PDU) has the same format as the 'Level 2 complete sequence numbers PDU' (Section 9.11), except for the PDU Type. The PDU Types for Levels 3 through 8 are defined as follows:

Level 3 (L3-CSNP-PDU): HHH

Level 4 (L4-CSNP-PDU): III

Level 5 (L5-CSNP-PDU): JJJ

Level 6 (L6-CSNP-PDU): KKK

Level 7 (L7-CSNP-PDU): LLL

Level 8 (L8-CSNP-PDU): MMM

### 3.5. Level n partial sequence numbers PDU (Ln-PSNP-PDU)

The 'Level 2 partial sequence numbers PDU' (Ln-PSNP-PDU) has the same format as the 'Level 2 partial sequence numbers PDU' (Section 9.13), except for the PDU Type. The PDU Types for Levels 3 through 8 are defined as follows:

Level 3 (L3-PSNP-PDU): NNN

Level 4 (L4-PSNP-PDU): OOO

Level 5 (L5-PSNP-PDU): PPP

Level 6 (L6-PSNP-PDU): QQQ

Level 7 (L7-PSNP-PDU): RRR

Level 8 (L8-PSNP-PDU): SSS

## 4. 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.

## 5. Acknowledgements

The author would like to thank Dinesh Dutt for inspiring this document.

## 6. IANA Considerations

This document makes many requests to IANA, as follows:

### 6.1. 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.

### 6.2. New PDUs

IANA is requested to allocate values from the IS-IS PDU registry for the following:

LAN-HELLO-PDU: AAA

L3-LSP-PDU: BBB

L4-LSP-PDU: CCC

L5-LSP-PDU: DDD

L6-LSP-PDU: EEE

L7-LSP-PDU: FFF

L8-LSP-PDU: GGG

L3-CSNP-PDU: HHH

L4-CSNP-PDU: III

L5-CSNP-PDU: JJJ

L6-CSNP-PDU: KKK

L7-CSNP-PDU: LLL

L8-CSNP-PDU: MMM

L3-PSNP-PDU: NNN

L4-PSNP-PDU: OOO

L5-PSNP-PDU: PPP

L6-PSNP-PDU: QQQ

L7-PSNP-PDU: RRR

L8-PSNP-PDU: SSS

To allow for PDU types to be defined independent of this document, the above values should be allocated from the range 32-254.

## 7. 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.

## 8. 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, 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>>.

## Author's Address

Tony Li  
Arista Networks  
5453 Great America Parkway  
Santa Clara, California 95054  
USA

Email: [tony.li@tony.li](mailto:tony.li@tony.li)

Internet Engineering Task Force  
Internet-Draft  
Intended status: Standards Track  
Expires: February 29, 2020

T. Li  
Arista Networks  
August 28, 2019

Area Abstraction for IS-IS  
draft-li-lsr-isis-area-abstraction-01

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, 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.

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 February 29, 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 Abstraction . . . . .	3
2.1. Area Leader Election . . . . .	4
2.2. LSP Generation . . . . .	5
2.3. Redundancy . . . . .	5
2.4. Level 2 SPF Considerations . . . . .	5
3. Area Proxy System Identifier TLV . . . . .	6
4. Acknowledgements . . . . .	6
5. IANA Considerations . . . . .	6
6. Security Considerations . . . . .	6
7. References . . . . .	7
7.1. Normative References . . . . .	7
7.2. Informative References . . . . .	7
Author's Address . . . . .	7

## 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 is protected.

This works well if the Level 1 area is tangential to the Level 2 area. This also works well if there are a number of 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 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 link state database. This implies that all routers that could possibly 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 onerous.

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 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 to 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 link state database held in the L3LS topology and eliminates any benefits of the hierarchy.

### 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. Area Abstraction

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 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 the purposes of this discussion, we'll consider a single Level 1 IS-IS area as the Target Area. All routers within this area speak Level 1 IS-IS on all of the links within this topology. We assume that the Target Area is always connected. We propose to implement Area Abstraction by having a Level 2 Proxy LSP that represents the

entire Target Area. This is the only LSP from the area that will be injected into the overall Level 2 link state database.

There are four classes of routers that we need to be concerned with in this discussion:

**Target Area Router** A router within the Target Area that runs Level 1 IS-IS. Some Target Area Routers may also run Level 2.

**Area Leader** The Area Leader is a Target Area Router that is elected to represent the Level 1 area by injecting the Proxy LSP into the Level 2 link state database. The Area Leader runs Level 2 as well as Level 1. There may be multiple candidates for Area Leader, but only one is elected at a given time.

**Area Edge Router** An Area Edge Router is a Target Area Router that also runs Level 2 and has at least one Level 2 interface outside of the Target Area.

**Area Neighbor** An Area Neighbor is a Level 2 router that is outside of the Target Area that has an adjacency with an Area Edge Router.

The Area Leader has several responsibilities. First, it must inject Area Proxy System Identifier into the Level 1 link state database. Second, the Area Leader must generate the Proxy LSP for the Target Area.

All Area Edge Routers learn the Area Proxy System Identifier from the Level 1 link state database and use that as the system identifier in their Level 2 IS-IS Hello PDUs on interfaces outside the Target Area. Area Neighbors should then advertise an adjacency to the Area Proxy System Identifier. The Area Edge Routers **MUST** also maintain a Level 2 adjacency with the Area Leader, either via a direct link or via a tunnel.

Area Edge Routers **MUST** be able to provide transit to Level 2 traffic. We propose that the Area Edge Routers use Segment Routing (SR) [I-D.ietf-spring-segment-routing] and, during Level 2 SPF computation, use the SR forwarding path to reach the exit Area Edge Routers. To support SR, Area Edge Routers **SHOULD** advertise Adjacency Segment Identifiers for their adjacency to the Area Leader. Other mechanisms are possible and are a local decision.

## 2.1. Area Leader Election

The Area Leader is selected using the election mechanisms described in Dynamic Flooding for IS-IS [I-D.ietf-lsr-dynamic-flooding].

## 2.2. LSP Generation

Each Area Edge Router generates a Level 2 LSP that includes adjacencies to any Area Neighbors and the Area Leader. Unlike normal Level 2 operations, this LSP is not advertised outside of the Target Area and must be filtered by all Area Edge Routers to not be flooded outside of the Target Area.

The Area Leader uses the Level 2 LSPs generated by the Area 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 Area Neighbors that have been advertised by the Area Edge Routers. Since the Area Neighbors also advertise an adjacency to the 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 link state database, with all other Level 2 LSPs from the Target Area being filtered out at the Target Area boundary.

## 2.3. 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.

## 2.4. Level 2 SPF Considerations

When Level 2 systems outside of the Target Area perform an Level 2 SPF computation, they will use the Area Proxy LSP for computing a path transiting the Target Area. Because the Level 1 topology has been abstracted away, the cost for transiting the Target Area will be zero.

When Level 2 systems inside of the Target Area perform a Level 2 computation, they must ignore the Area Proxy LSP. Further, because these systems do see the topology inside of the Target Area, the costs 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 Level 2 systems within the Target Area must consider the metrics of links outside of the Target Area (inter-area metrics) separately from the metrics of links inside of the Target Area (intra-area metrics). Intra-area metrics as being 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.

### 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: 2 + (length of a system ID)

Proxy System Identifier: The area's Proxy System Identifier, which is the length of a system identifier. field.

### 4. Acknowledgements

The author would like to thank Bruno Decraene for his many helpful comments. The author would also like to thank a small group that wishes to remain anonymous for their valuable contributions.

### 5. 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.

### 6. 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.

## 7. References

### 7.1. 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.
- [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.
- [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>>.

### 7.2. 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>>.

### Author's Address

Tony Li  
Arista Networks  
5453 Great America Parkway  
Santa Clara, California 95054  
USA

Email: [tony.li@tony.li](mailto:tony.li@tony.li)

Networking Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: August 29, 2019

Shaofu. Peng  
Ran. Chen  
Gregory. Mirsky  
ZTE Corporation  
February 25, 2019

Packet Network Slicing using Segment Routing  
draft-peng-lsr-network-slicing-00

Abstract

This document presents a mechanism aimed at providing a solution for network slicing in the transport network for 5G services. The proposed mechanism uses a unified administrative instance identifier to distinguish different virtual network resources for both intra-domain and inter-domain network slicing scenarios. Combined with the segment routing technology, the mechanism could be used for both best-effort and traffic engineered services for tenants.

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 August 29, 2019.

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. Conventions used in this document . . . . .	3
3. Overview of Mechanism . . . . .	3
4. Resource Allocation per AII . . . . .	5
4.1. L3 Link Resource AII Configuration . . . . .	5
4.2. L2 Link Resource AII Configuration . . . . .	5
4.3. Node Resource AII Configuration . . . . .	6
5. Interworking with SR Flex-algorithm . . . . .	6
5.1. Best-effort Service AII-specific . . . . .	7
5.2. Traffic Engineering service AII-specific . . . . .	7
6. Examples . . . . .	7
6.1. intra-domain network slicing . . . . .	8
6.2. inter-domain network slicing via BGP-LS . . . . .	9
6.3. inter-domain network slicing via BGP-LU . . . . .	11
7. Implementation suggestions . . . . .	11
8. IANA Considerations . . . . .	12
9. Security Considerations . . . . .	12
10. Acknowledgements . . . . .	12
11. Normative references . . . . .	12
Authors' Addresses . . . . .	13

## 1. Introduction

According to 5G context, network slicing is the collection of a set of technologies to create specialized, dedicated logical networks as a service (NaaS) in support of network service differentiation and meeting the diversified requirements from vertical industries. Through the flexible and customized design of functions, isolation mechanisms, and operation and management (O&M) tools, network slicing is capable of providing dedicated virtual networks over a shared infrastructure. A Network slice instance (NSI) is the realization of network slicing concept. It is an E2E logical network, which comprises of a group of network functions, resources, and connection relationships. An NSI typically covers multiple technical domains, which includes a terminal, access network (AN), transport network (TN) and a core network (CN), as well as DC domain that hosts third-party applications from vertical industries. Different NSIs may have different network functions and resources. They may also share some of the network functions and resources.

For a packet network, network slicing requires the underlying network to support partitioning of the network resources to provide the

client with dedicated (private) networking, computing, and storage resources drawn from a shared pool. The slices may be seen as virtual networks. [I-D.ietf-teas-enhanced-vpn] described a framework to create virtual networks in a packet network. This document specifies a detailed mechanism to signal association of shared resources required to create and manage an NSI.

Currently there are multiple methods that could be used to identify the virtual network resource, such as Administrative Group (AG) described in [RFC3630], [RFC5329] and [RFC5305], Extended Administrative Groups (EAGs) described in [RFC7308], and Multi-Topology Routing (MTR) described in [RFC5120], [RFC4915] and [RFC5340]. However, all these methods are not sufficient to meet the requirements of network slicing service. For example, AG or EAG are limited to serve as a link color scheme used in TE path computation to meet the requirements of TE service for a tenant. But it is difficult to use them for an NSI allocation mapping (assuming that each bit position of AG/EAG represents an NSI) and, at the same time, as an IGP FIB identifier for best-effort service for the same tenant. MTR is limited to serve as an IGP logical topology scheme only used in the intra-domain scenario, and it is challenging to select inter-area link resource according to MT-ID when E2E inter-domain TE path needs to be created for a tenant.

Thus, there needs to be a new characteristic of NSI to isolate underlay resources. Firstly it could serve as TE criteria for TE service, and secondly, as an IGP FIB table identifier for best-effort service. This document introduces a new property of NSI called "Administrative Instance Identifier" (AII) and corresponding method of using it.

## 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. Overview of Mechanism

At the initial stage, each link in a physical network can be colored to conform with network slicing requirements. As previously mentioned, AII can be used to color links to partition underlay resource. Also, we may continue to use AG or EAG to color links for traditional TE purpose within a virtual network specified by an AII. A single or multiple AIIs could be configured on each intra-domain or inter-domain link regardless of IGP instance configuration. At the minimum, a link always belongs to default AII (the value is 0). The number of AIIs configured on a node's links determines the number of



virtual networks the node is part of. This document defines a new extension of the existing IGP-TE mechanisms [RFC3630] and [RFC5305] to distribute AII information in an AS as a new TE parameter of a link. An SDN controller, using BGP-LS or another interface, will have a distinct view of each virtual network specified by AII.

Using the CSPF algorithm, a TE path for any best-effort (BE) or traffic engineered (TE) service can be calculated within a virtual network specified by the AII. The computation criteria could be <AII, min igp-metric> or <AII, traditional TE criterias> for the BE and TE respectively. Combined with segment routing, the TE path could be represented as:

- o a single node-SID of the destination node, for the best-effort service in the domain;
- o node-SIDs of the border node and the destination node, adjacency-SID of inter-domain link, for the inter-domain best-effort service;
- o an adjacency-SID list, for P2P traffic engineered service.

Because packets of the best-effort service could be transported over an MP2P LSP without congestion control, SR best-effort FIB for each virtual network specified by AII to forward best-effort packets may be created in the IGP domain. Thus, CSPF computation with criteria <AII, min igp-metric> is distributed on each node in the IGP domain. That is similar to the behavior in [Flex-algo], but the distributed CSPF computation is triggered by AII.

To distinguish forwarding behavior of different virtual networks, prefix-SID need to be allocated per AII and advertised in the IGP domain.

For inter-domain case, in addition to the destination node-SID, several node-SIDs of the domain border node and adjacency-SID of inter-domain link are also needed to construct the E2E segment list. The segment list could be computed with the help of the SDN controller which needs to consider AII information during the computation. The head-end of the segment list maintains the corresponding SR-TE tunnel or [I-D.ietf-spring-segment-routing-policy].

As for the prefix-SID, adjacency-SID needs to be allocated per AII to distinguish the forwarding behavior of different virtual networks.

For P2P traffic engineering service, especially such as uRLLC service, it SHOULD not transfer over an MP2P LSP to avoid the risk of

traffic congestion. The segment list could consist of pure adjacency-SID per AII specific. The head-end of the segment list maintains the corresponding SR-TE tunnel or [I-D.ietf-spring-segment-routing-policy].

However, label stack depth of the segment list MAY be optimized at a later time based on local policies.

At this moment we can steer traffic of overlay service to the above SR best-effort FIB, SR-TE tunnel or SR policy instance, for the specific virtual network. The overlay service could specify a color for TE purpose, for example, color 1000 means <AII=10, min igp-metric> to say that I need best-effort forwarding within AII 10 resource, color 1001 means <AII=10, delay=10ms, AG=0x1> to say that I need traffic engineering forwarding within AII 10 resource, especially using link with AG equal to 0x1 to reach guarantee of not exceeding 10ms delay time. Service with color 1000 will be steered to an SR best-effort FIB entry, or an SR-TE tunnel/policy in case of inter-domain. Service with color 1001 will be steered to an SR-TE tunnel/policy.

#### 4. Resource Allocation per AII

##### 4.1. L3 Link Resource AII Configuration

In IGP domain, each numbered or unnumbered L3 link could be configured with AII information and synchronized among IGP neighbors. The IGP link-state database will contain L3 links with AII information to support TE path computation considering AII criteria. For a numbered L3 link, it could be represented as a tuple <local node-id, remote node-id, local ip-address, remote ip-address>, for unnumbered it could be <local node-id, remote node-id, local interface-id, remote interface-id>. Each L3 link could be configured to belong to a single AII or multiple AII, for each <L3 link, AII> tuple it would allocate a different adjacency-SID. Note that an L3 link always belongs to default AII(0).

An L3 link that is not part of the IGP domain, such as the special purpose for a static route, or an inter-domain link, can also be configured with AII information and allocate adjacency-SID per AII as the same as IGP links. BGP-LS could be used to collect link state data with AII information to the controller.

##### 4.2. L2 Link Resource AII Configuration

[I-D.ietf-isis-l2bundles] described how to encode adjacency-SID for each L2 member link of an L3 parent link. It is beneficial to deploy LAG or another virtual aggregation interface in network slicing

scenario. Between two nodes, the dedicated link resources belong to different virtual networks could be added or removed on demand, they are treated as L2 member links of a single L3 virtual interface. It is the single L3 virtual interface that needs to occupy IP resource and be part of the IGP instance. Creating a new slice-specific link on demand or removing the old one, is likely to affect some configurations.

Each L2 member link of an L3 parent link SUGGESTED to be configured to belong to a single AII, and different L2 member link will have different single AII configuration, with different adjacency-SID. Note that in this case, the L3 parent link belongs to default AII(0), but each L2 member link belongs to the specific non-default AII. Routing protocol control packets follow the L3 parent link of the L2 member link with the highest priority. At the same time, data packets that belong to the specific virtual network will pass along the L2 member link with the specific AII value.

TE path computation based on link-state database need view the detailed L2 members of an L3 adjacency to select the expected L2 link resource.

#### 4.3. Node Resource AII Configuration

For topology resource, each node needs to allocate node-SID per AII when it joins the related virtual network. All nodes in the IGP domain can run CSPF algorithm with criteria <AII, min IGP metric> to compute best-effort next-hop to any other destination nodes for a virtual network AII-specific, based on the link-state database that containing AII information so that SR best-effort FIB can be constructed for each AII.

An intra-domain overlay best-effort service belongs to a virtual network could directly match in the above SR best-effort FIB for the specific AII, while an inter-domain overlay best-effort service belongs to a virtual network could be over a segment list containing domain border node-SID and destination node-SID which could match in the above SR best-effort FIB for the specific AII.

#### 5. Interworking with SR Flex-algorithm

[I-D.ietf-lsr-flex-algo] introduced a mechanism to do label stack depth optimization for an SR policy in IGP domain part. As the color of SR policy defined a TE purpose, traditionally the headend or SDN controller will compute an expected TE path to meet that purpose. It is necessary to map a color (32 bits) to an FA-ID (8 bits) when SR flex-algorithm enabled for an SR policy, besides that, it is necessary to enable the FA-ID on each node that will join the same FA

plane manually. However, the FAD could copy the TE constraints contained in the color template. We need to consider the cost of losing the flexibility of color when executing the flex-algo optimization, and also consider the gap between P2P TE requirements and MP2P SR LSP capability, to reach the right balance when deciding which SR policy need optimization.

#### 5.1. Best-effort Service AII-specific

As described above, for best-effort service we have already constructed SR best-effort FIB per AII, that is mostly like [Flex-algo]. Thus, it is not necessary to map to FA-ID again for a color template which has defined a best-effort behavior within the dedicated AII. Of course, if someone forced to remap it, there is no downside for the operation, the overlay best-effort service (with a color which defined specific AII, best-effort requirement, and mapping FA-ID) in IGP domain will try to recurse over <AII, prefix> or <FA-ID, prefix> FIB entry.

#### 5.2. Traffic Engineering service AII-specific

An SR-TE tunnel/policy that served for traffic engineering service of a virtual network specified by an AII was generated and computed according to the relevant color template, which contained specific AII and some other traditional TE constraints. If we config mapping FA-ID under the color template, the SR-TE tunnel/policy instance could inherit forwarding information from corresponding SR Flex-Algo FIB entry.

### 6. Examples

In this section, we will further illustrate the point through some examples. All examples share the same figure below.

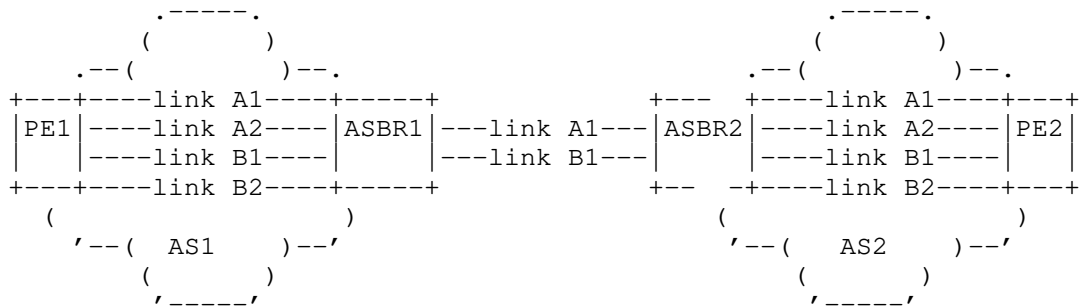


Figure 1 Network Slicing via AII

Suppose that each link belongs to separate virtual network, e.g., link Ax belongs to the virtual network colored by AII A, link Bx belongs to the virtual network colored by AII B. link x1 has an IGP metric smaller than link x2, but TE metric larger.

To simplify the use case, each AS just contained a single IGP area.

#### 6.1. intra-domain network slicing

From the perspective of node PE1 in AS1, it will calculate best-effort forwarding entry for each AII instance (including default AII) to destinations in the same IGP area. For example:

For <AII=0, destination=ASBR1> entry, forwarding information could be ECMP during link A1 and link B1, with destination node-SID 100 for <AII=0, destination=ASBR1>.

For <AII=A, destination=ASBR1> entry, forwarding information could be link A1, with destination node-SID 200 for <AII=A, destination=ASBR1>.

For <AII=B, destination=ASBR1> entry, forwarding information could be link B1, with destination node-SID 300 for <AII=B, destination=ASBR1>.

It could also initiate an SR-TE instance (SR tunnel or SR policy) with the particular color template on PE1, PE1 is headend and ASBR1 is destination node. For example:

For SR-TE instance 1 with color template which defined criteria including {default AII, min TE metric}, forwarding information could be ECMP during two segment list {adjacency-SID 1002 for <AII=0, link A2>@PE1} and {adjacency-SID 1004 for <AII=0, link B2>@PE1}.

For SR-TE instance 2 with the color template which defined criteria including {AII=A, min TE metric}, forwarding information could be presented as the segment list {adjacency-SID 2002 for <AII=A, link A2>@PE1}.

For SR-TE instance 3 with the color template which defined criteria including {AII=B, min TE metric}, forwarding information could be presented as the segment list {adjacency-SID 3004 for <AII=B, link B2>@PE1}.

Furthermore, we can use SR Flex-algo to optimize the above SR-TE instance. For example, for SR-TE instance 1, we can define FA-ID 201 with FAD that contains the same information as the color template, in turn, FA-ID 202 for SR-TE instance 2, FA-ID 203 for SR-TE instance 3. Note that each FA-ID also needs to be enabled on ASBR1. So that the corresponding SR FA entry could be:

For <FA-ID=201, destination=ASBR1> entry, forwarding information could be ECMP during link A2 and link B2, with destination node-SID 600 for <FA-ID=201, destination=ASBR1>.

For <FA-ID=202, destination=ASBR1> entry, forwarding information could be link A2, with destination node-SID 700 for <FA-ID=202, destination=ASBR1>.

For <FA-ID=203, destination=ASBR1> entry, forwarding information could be link B2, with destination node-SID 800 for <FA-ID=203, destination=ASBR1>.

## 6.2. inter-domain network slicing via BGP-LS

An E2E inner-AS SR-TE instance with particular color template could be initiated on PE1, PE1 is head-end and PE2 is destination node. BGP-LS could be used to inform the SDN controller about the underlay network topology information including AII attribute. Thus the controller could calculate E2E TE path within the particular virtual network. For best-effort service, for example:

For SR-TE instance 4 with color template which defined criteria including {default AII, min IGP metric}, forwarding information could be segment list {node-SID 100 for <AII=0, destination=ASBR1>, adjacency-SID 1001 for <AII=0, link A1>@ASBR1, node-SID 400 for <AII=0, destination=PE2>}.

For SR-TE instance 5 with color template which defined criteria including {AII=A, min IGP metric}, forwarding information could be segment list {node-SID 200 for <AII=A, destination=ASBR1>, adjacency-

SID 1001 for <AII=A, link A1>@ASBR1, node-SID 500 for <AII=A, destination=PE2>}

For SR-TE instance 6 with color template which defined criteria including {AII=B, min IGP metric}, forwarding information could be segment list {node-SID 300 for <AII=B, destination=ASBR1>, adjacency-SID 1003 for <AII=B, link B1>@ASBR1, node-SID 600 for <AII=B, destination=PE2>}

For TE service, for example:

For SR-TE instance 7 with color template which defined criteria including {default AII, min TE metric}, forwarding information could be ECMP during two segment list {adjacency-SID 1002 for <AII=0, link A2>@PE1, adjacency-SID 1001 for <AII=0, link A1>@ASBR1, adjacency-SID 1002 for <AII=0, link A2>@ASBR2} and {adjacency-SID 1004 for <AII=0, link B2>@PE1, adjacency-SID 1003 for <AII=0, link B1>@ASBR1, adjacency-SID 1004 for <AII=0, link B2>@ASBR2}

For SR-TE instance 8 with color template which defined criteria including {AII=A, min TE metric}, forwarding information could be segment list {adjacency-SID 2002 for <AII=A, link A2>@PE1, adjacency-SID 2001 for <AII=A, link A1>@ASBR1, adjacency-SID 2002 for <AII=A, link A2>@ASBR2}

For SR-TE instance 9 with color template which defined criteria including {AII=B, min TE metric}, forwarding information could be segment list {adjacency-SID 3004 for <AII=B, link B2>@PE1, adjacency-SID 3003 for <AII=B, link B1>@ASBR1, adjacency-SID 3004 for <AII=B, link B2>@ASBR2}

For TE service, if we use SR Flex-algo to do optimization, the above forwarding information of each TE instance could inherit the corresponding SR FA entry, it would look like this:

For SR-TE instance 7, forwarding information could be ECMP during two segment list {node-SID 600 for <FA-ID=201, destination=ASBR1>, adjacency-SID 1001 for <AII=0, link A1>@ASBR1, node-SID 600 for <FA-ID=201, destination=PE2>} and {adjacency-SID 1004 for <AII=0, link B2>@PE1, adjacency-SID 1003 for <AII=0, link B1>@ASBR1, adjacency-SID 1004 for <AII=0, link B2>@ASBR2}

For SR-TE instance 8 with color template which defined criteria including {AII=A, min TE metric}, forwarding information could be segment list {node-SID 700 for <FA-ID=202, destination=ASBR1>, adjacency-SID 2001 for <AII=A, link A1>@ASBR1, node-SID 700 for <FA-ID=202, destination=PE2>}

For SR-TE instance 9 with color template which defined criteria including {AII=B, min TE metric}, forwarding information could be segment list {node-SID 800 for <FA-ID=203, destination=ASBR1>, adjacency-SID 3003 for <AII=B, link B1>@ASBR1, node-SID 800 for <FA-ID=203, destination=PE2>}.

### 6.3. inter-domain network slicing via BGP-LU

In some deployments, operators adopt BGP-LU to build inter-domain MPLS LSP, overlay service will be directly over BGP-LU LSP. If overlay service has TE requirements that defined by a color, that means that BGP-LU LSP needs to have a sense of color too, i.e., BGP-LU label could be allocated per color. BGP-LU LSP generated for specific color will be over intra-domain SR-TE or SR Best-effort path generated for that color again.

In figure 1, PE2 can allocate and advertise six labels for its loopback plus color 1, 2, 3, 4, 5, 6 respectively. Suppose color 1 defines {default AII, min IGP metric}, color 2 defines {AII=A, min IGP metric}, color 3 defines {AII=B, min IGP metric}, and color 4 defines {default AII, min TE metric}, color 5 defines {AII=A, min TE metric}, color 6 defines {AII=B, min TE metric}. PE2 will advertise these labels to ASBR2 and ASBR2 then continues to allocate six labels each for prefix PE2 plus different color. Other nodes will have the same operation. Ultimately PE1 will maintain six BGP-LU LSP.

For example, the BGP-LU LSP for color 1 will be over SR best-effort FIB entry node-SID 100 for <AII=0, destination=ASBR1> to pass through AS1, over adjacency-SID 1001 for <AII=0, link A1>@ASBR1 to pass inter-AS, over SR best-effort FIB entry node-SID 400 for <AII=0, destination=PE2> to pass through AS2.

For example, The BGP-LU LSP for color 4 will over SR-TE instance 1 (see section 6.1), or SR best-effort FIB entry node-SID 600 for <FA-ID=201, destination=ASBR1> (see section 6.1) to pass through 6AS1, over adjacency-SID 1001 for <AII=0, link A1>@ASBR1 to pass inter-AS, over SR-TE instance 1' or corresponding SR FA entry to pass through AS2.

## 7. Implementation suggestions

As a node often contains control plane and forwarding plane, a suggestion is that only default AII specific FTN entry need be installed on forwarding plane, so that there are not any modification and upgrade requirement for hardware and existing MPLS forwarding mechanism. FTN entry for non-default AII instance will only be maintained on the control plane and be used for overlay service iteration according to next-hop plus color (color will give AII



information and mapping FA-ID information). Note ILM entry for all AII need be installed on forwarding plane.

The same suggestion is also appropriate for SR Flex-algo.

## 8. IANA Considerations

TBD.

## 9. Security Considerations

TBD.

## 10. Acknowledgements

TBD.

## 11. Normative references

[I-D.ietf-isis-l2bundles]

Ginsberg, L., Bashandy, A., Filsfils, C., Nanduri, M., and E. Aries, "Advertising L2 Bundle Member Link Attributes in IS-IS", draft-ietf-isis-l2bundles-07 (work in progress), May 2017.

[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-01 (work in progress), November 2018.

[I-D.ietf-spring-segment-routing-policy]

Filsfils, C., Sivabalan, S., daniel.voyer@bell.ca, d., bogdanov@google.com, b., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-02 (work in progress), October 2018.

[I-D.ietf-teas-enhanced-vpn]

Dong, J., Bryant, S., Li, Z., Miyasaka, T., and Y. Lee, "A Framework for Enhanced Virtual Private Networks (VPN+) Service", draft-ietf-teas-enhanced-vpn-01 (work in progress), February 2019.

[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>>.

- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.

## Authors' Addresses

Shaofu Peng  
ZTE Corporation

Email: [peng.shaofu@zte.com.cn](mailto:peng.shaofu@zte.com.cn)

Ran Chen  
ZTE Corporation

Email: [chen.ran@zte.com.cn](mailto:chen.ran@zte.com.cn)

Gregory Mirsky  
ZTE Corporation

Email: gregimirsky@gmail.com

Network Work group  
Internet-Draft  
Intended status: Standards Track  
Expires: May 18, 2019

P. Psenak, Ed.  
N. Nainar, Ed.  
IJ. Wijnands  
Cisco Systems, Inc.  
November 14, 2018

OSPFv3 Extensions for BIER  
draft-psenak-bier-ospfv3-extensions-02

Abstract

Bit Index Explicit Replication (BIER) is an architecture that provides multicast forwarding through a "BIER domain" without requiring intermediate routers to maintain multicast related per-flow state. Neither does BIER require an explicit tree-building protocol for its operation. A multicast data packet enters a BIER domain at a "Bit-Forwarding Ingress Router" (BFIR), and leaves the BIER domain at one or more "Bit-Forwarding Egress Routers" (BFERs). The BFIR router adds a BIER header to the packet. Such header contains a bit-string in which each bit represents exactly one BFER to forward the packet to. The set of BFERs to which the multicast packet needs to be forwarded is expressed by the according set of bits set in BIER packet header.

This document describes the OSPFv3 [RFC8362] protocol extensions required for BIER with MPLS encapsulation [RFC8296]. Support for other encapsulation types is outside the scope of this document. The use of multiple encapsulation types is outside the scope of 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 May 18, 2019.

## Copyright Notice

Copyright (c) 2018 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. Flooding of the BIER Information in OSPFv3 . . . . .	3
2.1. BIER Sub-TLV . . . . .	3
2.2. BIER MPLS Encapsulation Sub-TLV . . . . .	5
2.3. Flooding scope of BIER Information . . . . .	6
3. Security Considerations . . . . .	7
4. IANA Considerations . . . . .	8
5. Acknowledgements . . . . .	9
6. Normative References . . . . .	9
Authors' Addresses . . . . .	10

## 1. Introduction

Bit Index Explicit Replication (BIER) is an architecture that provides optimal multicast forwarding through a "BIER domain" without requiring intermediate routers to maintain any multicast related per-flow state. Neither does BIER explicitly require a tree-building protocol for its operation. A multicast data packet enters a BIER domain at a "Bit-Forwarding Ingress Router" (BFIR), and leaves the BIER domain at one or more "Bit-Forwarding Egress Routers" (BFERs). The BFIR router adds a BIER header to the packet. The BIER header contains a bit-string in which each bit represents exactly one BFER to forward the packet to. The set of BFERs to which the multicast packet needs to be forwarded is expressed by setting the bits that correspond to those routers in the BIER header.

BIER architecture requires routers participating in BIER to exchange BIER related information within a given domain. BIER architecture permits link-state routing protocols to perform distribution of such information. [RFC8444] proposes the OSPFv2 protocol extensions to distribute BIER specific information. This document describes

extensions to OSPFv3 necessary to advertise BIER specific information in the case where BIER uses MPLS encapsulation as described in [RFC8296].

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. Flooding of the BIER Information in OSPFv3

All BIER specific information that a Bit-Forwarding Router (BFR) needs to advertise to other BFRs is associated with a BFR-Prefix. A BFR prefix is a unique (within a given BIER domain) routable IPv4 or IPv6 address that is assigned to each BFR as described in more detail in [RFC8279].

[RFC8362] defines the encoding of OSPFv3 LSA in TLV format that allows to carry additional informations. This section defines the required Sub-TLVs to carry BIER information that is associated with the BFR-Prefix. The Sub-TLV defined in this section MAY be carried in the below OSPFv3 Extended LSA TLVs [RFC8362]:

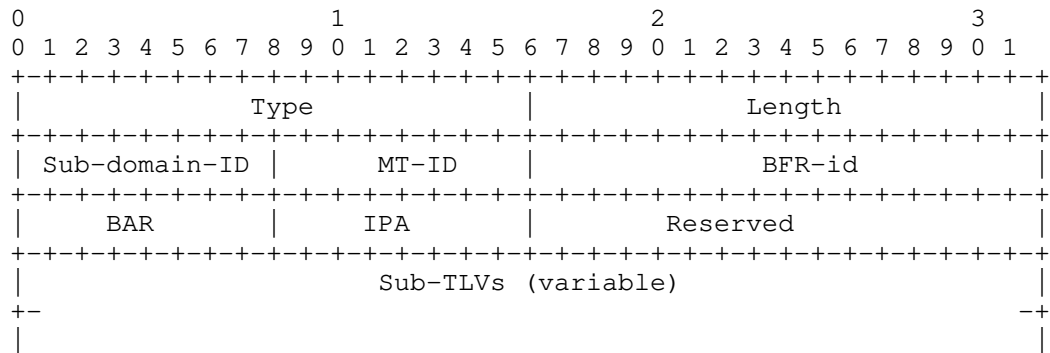
Intra-Area-Prefix TLV

Inter-Area-Prefix TLV

### 2.1. BIER Sub-TLV

A Sub-TLV of the above mentioned Prefix TLVs is defined for distributing BIER information. The Sub-TLV is called the BIER Sub-TLV. Multiple BIER Sub-TLVs may be included in any of the above mentioned Prefix TLV.

The BIER Sub-TLV has the following format:



Type: TBD1

Length: Variable, dependent on sub-TLVs.

Sub-domain-ID: Unique value identifying the BIER sub-domain within the BIER domain, as described in [RFC8279]

MT-ID: Multi-Topology ID (as defined in [RFC4915] that identifies the topology that is associated with the BIER sub-domain.

BFR-id: A 2 octet field encoding the BFR-id, as documented in section 2 of [RFC8279]. If the BFR is not locally configured with a valid BFR-id, the value of this field is set to 0, which is defined as illegal in [RFC8279].

BAR: Single octet BIER specific algorithm used to calculate underlay paths to reach other BFRs. Values are allocated from the "BIER Algorithm" registry which is defined in [RFC8401].

IPA: Single octet IGP algorithm to either modify, enhance or replace the calculation of underlay paths to reach other BFRs as defined by the BAR value. Values are defined in the "IGP Algorithm Types" registry.

Each BFR sub-domain MUST be associated with one and only one OSPF topology that is identified by the MT-ID. If the association between BIER sub-domain and OSPF topology advertised in the BIER sub-TLV by other BFRs is in conflict with the association locally configured on the receiving router, the BIER Sub-TLV MUST be ignored.

If the MT-ID value is outside of the values specified in [RFC4915], the BIER Sub-TLV MUST be ignored.

If a BFR advertises the same Sub-domain-ID in multiple BIER sub-TLVs, the BFR MUST be treated as if it did not advertise a BIER sub-TLV for such sub-domain.

All BFRs MUST detect advertisement of duplicate valid BFR-IDs for a given MT-ID and Sub-domain-ID. When such duplication is detected by the BFR, it MUST behave as described in section 5 of [RFC8279].

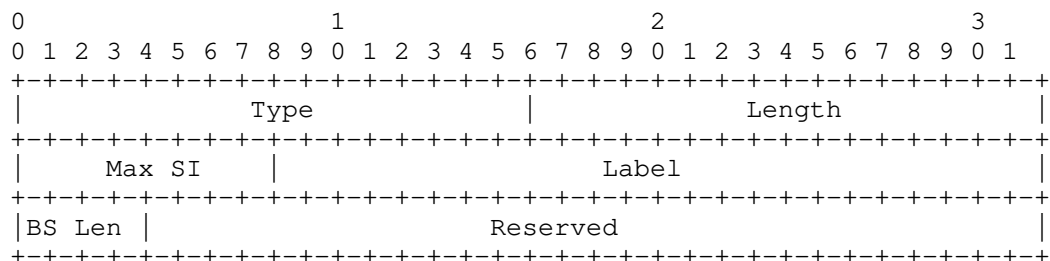
The supported BAR and IPA algorithms MUST be consistent for all routers supporting a given BFR sub-domain. A router receiving BIER Sub-TLV advertisement with a value in BAR or IPA fields which does not match the locally configured value for a given BFR sub-domain, MUST report a misconfiguration for such BIER sub-domain and MUST ignore such BIER sub-TLV.

The use of non-zero values in either the BAR field or the IPA field is outside the scope of this document.

## 2.2. BIER MPLS Encapsulation Sub-TLV

The BIER MPLS Encapsulation Sub-TLV is a Sub-TLV of the BIER Sub-TLV defined in Section 2.1. The BIER MPLS Encapsulation Sub-TLV is used in order to advertise MPLS specific information used for BIER. It MAY appear multiple times in the BIER Sub-TLV.

The BIER MPLS Encapsulation Sub-TLV has the following format:



Type: Set to TBD2.

Length: 8 octets

Max SI: A 1 octet field encoding the maximum Set Identifier (section 1 of [RFC8279]), used in the encapsulation for this BIER sub-domain for this bitstring length.



**Label:** A 3 octet field, where the 20 rightmost bits represent the first label in the label range. The 4 leftmost bits MUST be ignored.

**Bit String Length:** A 4 bits field encoding the supported BitString length associated with this BFR-prefix. The values allowed in this field are specified in section 2 of [RFC8296].

**Reserved:** SHOULD be set to 0 on transmission and MUST be ignored on reception.

The "label range" is the set of labels beginning with the Label and ending with (Label + (Max SI)). A unique label range is allocated for each BitString length and Sub-domain-ID. These labels are used for BIER forwarding as described in [RFC8279] and [RFC8296].

The size of the label range is determined by the number of Set Identifiers (SI) (section 1 of [RFC8279]) that are used in the network. Each SI maps to a single label in the label range. The first label is for SI=0, the second label is for SI=1, etc.

If the label associated with the Maximum Set Identifier exceeds the 20 bit range, the BIER MPLS Encapsulation Sub-TLV MUST be ignored.

If the BS length is set to a value that does not match any of the allowed values specified in [RFC8296], the BIER MPLS Encapsulation Sub-TLV MUST be ignored.

If same BS length is repeated in multiple BIER MPLS Encapsulation Sub-TLV inside the same BIER Sub-TLV, the BIER sub-TLV MUST be ignored.

Label ranges within all BIER MPLS Encapsulation Sub-TLVs advertised by the same BFR MUST NOT overlap. If the overlap is detected, the advertising router MUST be treated as if it did not advertise any BIER sub-TLVs.

### 2.3. Flooding scope of BIER Information

The flooding scope of the Extended LSAs [RFC8362] that is used for advertising the BIER Sub-TLV is area-local. To allow BIER deployment in a multi-area environment, OSPFv3 must propagate BIER information between areas.

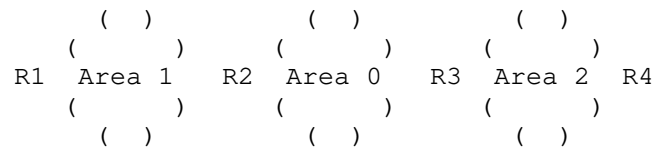


Figure 1: BIER propagation between areas

The following procedure is used in order to propagate BIER related information between areas:

When an OSPFv3 Area Border Router (ABR) advertises E-Inter-Area-Prefix-LSA from an intra-area or inter-area prefix to all its attached areas, it determines whether a BIER Sub-TLV should be included in this LSA. When doing so, an OSPFv3 ABR will:

- \* Examine its best path to the prefix in the source area and find the advertising router associated with the best path to that prefix.
- \* Determine if such advertising router advertised a BIER Sub-TLV for the prefix. If yes, the ABR will copy the information from such BIER Sub-TLV when advertising BIER Sub-TLV to each attached area.

In the Figure 1, R1 advertises a prefix 2001:db8:ble6::1/128 in Area 1. It also includes BIER Sub-TLV in E-Intra-Area-Prefix-LSA. ABR R2 calculates the reachability for prefix 2001:bdb8:ble6::1/128 inside Area 1 and propagates it to Area 0 using E-Inter-Area-Prefix-LSA. When doing so, it copies the entire BIER Sub-TLV (including all its Sub-TLVs) it received from R1 in Area 1 and includes it in the E-Inter-Area-Prefix-LSA it generates for the prefix in Area 0. ABR R3 calculates the reachability for prefix 2001:bdb8:ble6::1/128 inside Area 0 and propagates it to Area 2. When doing so, it copies the entire BIER Sub-TLV (including all its Sub-TLVs) it received from R2 in Area 0 and includes it in E-Inter-Area-Prefix-LSA it generates for 2001:bdb8:ble6::1/128 in Area 2.

### 3. Security Considerations

This document introduces new sub-TLVs for OSPFv3 Extended-LSAs. It does not introduce any new security risks to OSPFv3. Existing security concerns documented in [RFC8362] is applicable for the Sub-TLVs defined in this document.

It is assumed that both BIER and OSPF layer is under a single administrative domain. There can be deployments where potential

attackers have access to one or more networks in the OSPFv3 routing domain. In these deployments, stronger authentication mechanisms such as those specified in [RFC4552] SHOULD be used.

The Security Considerations section of [RFC8279] discusses the possibility of performing a Denial of Service (DoS) attack by setting too many bits in the BitString of a BIER-encapsulated packet. However, this sort of DoS attack cannot be initiated by modifying the OSPF BIER advertisements specified in this document. A BFIR decides which systems are to receive a BIER-encapsulated packet. In making this decision, it is not influenced by the OSPF control messages. When creating the encapsulation, the BFIR sets one bit in the encapsulation for each destination system. The information in the OSPF BIER advertisements is used to construct the forwarding tables that map each bit in the encapsulation into a set of next hops for the host that is identified by that bit, but is not used by the BFIR to decide which bits to set. Hence an attack on the OSPF control plane cannot be used to cause this sort of DoS attack.

While a BIER-encapsulated packet is traversing the network, a BFR that receives a BIER-encapsulated packet with *n* bits set in its BitString may have to replicate the packet and forward multiple copies. However, a given bit will only be set in one copy of the packet. That means that each transmitted replica of a received packet has fewer bits set (i.e., is targeted to fewer destinations) than the received packet. This is an essential property of the BIER forwarding process as defined in [RFC8279]. While a failure of this process might cause a DoS attack (as discussed in the Security Considerations of [RFC8279]), such a failure cannot be caused by an attack on the OSPF control plane.

Implementations MUST assure that malformed TLV and Sub-TLV defined in this document are detected and do not provide a vulnerability for attackers to crash the OSPFv3 router or routing process. Reception of malformed TLV or Sub-TLV SHOULD be counted and/or logged for further analysis. Logging of malformed TLVs and Sub-TLVs SHOULD be rate-limited to prevent a Denial of Service (DoS) attack (distributed or otherwise) from overloading the OSPFv3 control plane.

#### 4. IANA Considerations

The document requests two new allocations from the OSPFv3 Extended-LSA sub-TLV registry as defined in [RFC8362].

BIER Sub-TLV: TBD1

BIER MPLS Encapsulation Sub-TLV: TBD2

## 5. Acknowledgements

TBD

## 6. 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>>.
- [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>>.
- [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>>.
- [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>>.
- [RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <<https://www.rfc-editor.org/info/rfc8279>>.
- [RFC8296] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Tantsura, J., Aldrin, S., and I. Meilik, "Encapsulation for Bit Index Explicit Replication (BIER) in MPLS and Non-MPLS Networks", RFC 8296, DOI 10.17487/RFC8296, January 2018, <<https://www.rfc-editor.org/info/rfc8296>>.
- [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>>.
- [RFC8401] Ginsberg, L., Ed., Przygienda, T., Aldrin, S., and Z. Zhang, "Bit Index Explicit Replication (BIER) Support via IS-IS", RFC 8401, DOI 10.17487/RFC8401, June 2018, <<https://www.rfc-editor.org/info/rfc8401>>.

[RFC8444] Psenak, P., Ed., Kumar, N., Wijnands, IJ., Dolganow, A., Przygienda, T., Zhang, J., and S. Aldrin, "OSPFv2 Extensions for Bit Index Explicit Replication (BIER)", RFC 8444, DOI 10.17487/RFC8444, November 2018, <<https://www.rfc-editor.org/info/rfc8444>>.

#### Authors' Addresses

Peter Psenak (editor)  
Cisco Systems, Inc.  
Apollo Business Center  
Mlynske nivy 43, Bratislava 821 09  
Slovakia

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

Nagendra Kumar Nainar (editor)  
Cisco Systems, Inc.  
7200 Kit Creek Road  
Research Triangle Park, NC 27709  
US

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

IJsbrand Wijnands  
Cisco Systems, Inc.  
De Kleetlaan 6a  
Diegem 1831  
Belgium

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