

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

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

OSPF YANG Model Augmentations for Additional Features - Version 1
draft-acee-lsr-ospf-yang-augmentation-v1-01

Abstract

This document defines YANG data modules augmenting the IETF OSPF YANG model to provide support for Traffic Engineering Extensions to OSPF Version 3 as defined in RF 5329, OSPF Two-Part Metric as defined in RFC 8042, OSPF Graceful Link Shutdown as defined in RFC 8379 and OSPF Link-Local Signaling (LLS) Extensions for Local Interface ID Advertisement as defined in RFC 8510.

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	3
2. YANG Module for Traffic Engineering Extensions to OSPF Version 3	3
3. YANG Module for OSPF Two-Part Metric	8
4. YANG Module for OSPF Graceful Link Shutdown	12
5. YANG Module for OSPF LLS Extension for Local Interface ID Advertisement	17
6. Security Considerations	19
7. IANA Considerations	20
8. Acknowledgements	21
9. References	21
9.1. Normative References	21
9.2. Informative References	22
Authors' Addresses	22

1. Overview

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

This document defines YANG data modules augmenting the IETF OSPF YANG model [I-D.ietf-ospf-yang], which itself augments [RFC8349], to provide support for configuration and operational state for the following OSPF features:

RFC5329: Traffic Engineering Extensions to OSPF Version 3 [RFC5329].

RFC8042: OSPF Two-Part Metric [RFC8042].

RFC8379: OSPF Graceful Link Shutdown [RFC8379].

RFC8510: OSPF Link-Local Signaling (LLS) Extensions for Local Interface ID Advertisement [RFC8510].

The augmentations defined in this document requires support for the OSPF base model [I-D.ietf-ospf-yang] which defines basic OSPF

configuration and state. The OSPF YANG model augments the ietf-routing YANG model defined in [RFC8022].

1.1. Requirements Language

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

2. YANG Module for Traffic Engineering Extensions to OSPF Version 3

This document defines a YANG module for Traffic Engineering Extensions to OSPF Version 3 as defined in [RFC5329]. It is an augmentation of the OSPF base model.

```
module: ietf-ospfv3-te
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
    /ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
    /ospf:area-scope-lsa/ospf:version/ospf:ospfv3/ospf:ospfv3
    /ospf:body:
  +--ro ospfv3-intra-area-te
    +--ro router-address-tlv
      | +--ro router-address?   inet:ipv6-address
    +--ro link-tlv
      +--ro link-type           ospf:router-link-type
      +--ro local-if-ipv6-addr
        | +--ro local-if-ipv6-addr*   inet:ipv6-address
      +--ro remote-if-ipv6-addr
        | +--ro remote-if-ipv6-addr*   inet:ipv6-address
      +--ro te-metric?          uint32
      +--ro max-bandwidth?      rt-types:bandwidth-ieee-float32
      +--ro max-reservable-bandwidth?  rt-types:bandwidth-ieee-float32
      +--ro unreserved-bandwidths
        | +--ro unreserved-bandwidth*
          | +--ro priority?          uint8
          | +--ro unreserved-bandwidth?  rt-types:bandwidth-ieee-float32
      +--ro admin-group?       uint32
      +--ro neighbor-id
        | +--ro nbr-interface-id   inet:ipv4-address
        | +--ro nbr-router-id     yang:dotted-quad
      +--ro unknown-tlvs
        +--ro unknown-tlv*
          +--ro type?            uint16
          +--ro length?         uint16
          +--ro value?          yang:hex-string
```

<CODE BEGINS> file "ietf-ospfv3-te@2019-08-13.yang"

```
module ietf-ospfv3-te {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-ospfv3-te";

  prefix ospfv3-te;

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

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

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

  import ietf-routing {
    prefix "rt";
  }

  import ietf-ospf {
    prefix "ospf";
  }

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

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

     Author:    Yingzhen Qu
                <mailto:yqu@futurewei.com>
     Author:    Acee Lindem
                <mailto:acee@cisco.com>";

  description
    "This YANG module defines the configuration and operational
     state for OSPFv3 extensions to support intra-area Traffic
     Engineering (TE) as defined in RFC 5329."

  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 version";
  reference
    "RFC XXXX: A YANG Data Model for OSPFv3 TE.";
}

identity ospfv3-intra-area-te-lsa {
  base ospf:ospfv3-lsa-type;
  description
    "OSPFv3 Intrea-area TE LSA.";
}

grouping ospfv3-intra-area-te {
  description "Grouping for OSPFv3 intra-area-te-lsa.";
  container ospfv3-intra-area-te {
    container router-address-tlv {
      description "The router IPv6 address tlv advertises a reachable
        IPv6 address.";
      leaf router-address {
        type inet:ipv6-address;
        description
          "Router IPv6 address.";
      }
    }
  }

  container link-tlv {
    description "Describes a singel link, and it is constructed
      of a set of Sub-TLVs.";
    leaf link-type {
      type ospf:router-link-type;
      mandatory true;
      description "Link type.";
    }
  }
}
```

```
container local-if-ipv6-addrs {
  description "All local interface IPv6 addresses.";
  leaf-list local-if-ipv6-addr {
    type inet:ipv6-address;
    description
      "List of local interface IPv6 addresses.";
  }
}

container remote-if-ipv6-addrs {
  description "All remote interface IPv6 addresses.";
  leaf-list remote-if-ipv6-addr {
    type inet:ipv6-address;
    description
      "List of remote interface IPv6 addresses.";
  }
}

leaf te-metric {
  type uint32;
  description "TE metric.";
}

leaf max-bandwidth {
  type rt-types:bandwidth-ieee-float32;
  description "Maximum bandwidth.";
}

leaf max-reservable-bandwidth {
  type rt-types:bandwidth-ieee-float32;
  description "Maximum reservable bandwidth.";
}

container unreserved-bandwidths {
  description "All unreserved bandwidths.";
  list unreserved-bandwidth {
    leaf priority {
      type uint8 {
        range "0 .. 7";
      }
      description "Priority from 0 to 7.";
    }
    leaf unreserved-bandwidth {
      type rt-types:bandwidth-ieee-float32;
      description "Unreserved bandwidth.";
    }
  }
  description
    "List of unreserved bandwidths for different
```

```

        priorities.";
    }
}

leaf admin-group {
    type uint32;
    description
        "Administrative group/Resource Class/Color.";
}

container neighbor-id {
    description "Neighbor link identification.";
    leaf nbr-interface-id {
        type inet:ipv4-address;
        mandatory true;
        description "The neighbor's interface ID.";
    }
    leaf nbr-router-id {
        type yang:dotted-quad;
        mandatory true;
        description "The neighbor's router ID.";
    }
}

uses ospf:unknown-tlvs;
}

description "OSPFv3 Intra-Area-TE-LSA.";
reference "RFC 5329: Traffic Engineering Extensions to OSPF
        :   Version 3.";
}
}

augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:areas/"
+ "ospf:area/ospf:database/"
+ "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
+ "ospf:area-scope-lsa/ospf:version/ospf:ospfv3/"
+ "ospf:ospfv3/ospf:body" {
when "../.../.../.../.../.../.../.../..."
+ "rt:type = 'ospf:ospfv3'" {
description
    "This augmentation is only valid for OSPFv3.";
}
description
    "OSPFv3 Intrea-Area-TE-LSA.";
}

```

```

    uses ospfv3-intra-area-te;
  }
}
<CODE ENDS>

```

3. YANG Module for OSPF Two-Part Metric

This document defines a YANG module for OSPF Two-Part Metric feature as defined in [RFC8042]. It is an augmentation of the OSPF base model.

```

module: ietf-ospf-two-part-metric
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
    /ospf:interfaces/ospf:interface:
    +--rw two-part-metric
      +--rw enable?          boolean
      +--rw int-input-cost?  ospf:ospf-link-metric
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
    /ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
    /ospf:area-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
    /ospf:body/ospf:opaque/ospf:extended-link-opaque
    /ospf:extended-link-tlv:
    +--ro network-to-router-metric-sub-tlvs
      +--ro net-to-rtr-sub-tlv*
        +--ro mt-id?        uint8
        +--ro mt-metric?    uint16
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
    /ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
    /ospf:area-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
    /ospf:body/ospf:opaque/ospf:te-opaque/ospf:link-tlv:
    +--ro network-to-router-te-metric?  uint32
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
    /ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
    /ospf:area-scope-lsa/ospf:version/ospf:ospfv3/ospf:ospfv3
    /ospf:body/ospfv3-te:ospfv3-intra-area-te/ospfv3-te:link-tlv:
    +--ro network-to-router-te-metric?  uint32

<CODE BEGINS> file "ietf-ospf-two-part-metric@2019-08-13.yang"
module ietf-ospf-two-part-metric {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-ospf-two-part-metric";

  prefix ospf-two-metric;

```



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

import ietf-ospf {
  prefix "ospf";
}

import ietf-ospfv3-te {
  prefix "ospfv3-te";
}

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

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

   Author:    Yingzhen Qu
               <mailto:yqu@futurewei.com>
   Author:    Acee Lindem
               <mailto:acee@cisco.com>";

description
  "This YANG module defines the configuration and operational
  state for OSPF Two-Part Metric feature as defined in RFC 8042.

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

```
}

identity two-part-metric {
  base ospf:informational-capability;
  description
    "When set, the router is capable of supporting OSPF
    two-part metrics.";
  reference
    "RFC 8042: OSPF Two-Part Metric";
}

/* RFC 8042 */
augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/ospf:ospf/"
  + "ospf:areas/ospf:area/ospf:interfaces/ospf:interface" {
  when "../.../rt:type = 'ospf:ospfv2' or "
  + "../.../rt:type = 'ospf:ospfv3'" {
    description
      "This augments the OSPF interface configuration
      when used.";
  }
  description
    "This augments the OSPF protocol interface
    configuration with two-part metric.";

  container two-part-metric {
    when "enum-value(../ospf:interface-type) = 2" {
      description
        "Two-part metric when link type is multi-access.";
    }
    leaf enable {
      type boolean;
      default false;
      description
        "Enable two-part metric.";
    }
    leaf int-input-cost {
      type ospf:ospf-link-metric;
      description
        "Link state metric from the two-part-metric network
        to this router.";
    }
  }
  description
    "Interface two part metric configuration.";
}
}
```

```
augment "/rt:routing/"
```

```

    + "rt:control-plane-protocols/rt:control-plane-protocol/"
    + "ospf:ospf/ospf:areas/"
    + "ospf:area/ospf:database/"
    + "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
    + "ospf:area-scope-lsa/ospf:version/ospf:ospfv2/"
    + "ospf:ospfv2/ospf:body/ospf:opaque/"
    + "ospf:extended-link-opaque/ospf:extended-link-tlv" {
when "../.../.../.../.../.../.../.../.../.../.../..."
    + "rt:type = 'ospf:ospfv2'" {
    description
        "This augmentation is only valid for OSPFv2.";
    }
description
    "Network-to-Router metric sub tlv for OSPFv2 extended link TLV
    in type 10 opaque LSA.";

    container network-to-router-metric-sub-tlvs {
        description "Network-to-Router metric sub TLV.";
        list net-to-rtr-sub-tlv {
            leaf mt-id {
                type uint8;
                description "Multi-Topology Identifier (MT-ID).";
            }
            leaf mt-metric {
                type uint16;
                description "Network-to-router metric.";
            }
        }
        description
            "Network-to-Router metric sub-TLV.";
    }
}

augment "/rt:routing/"
    + "rt:control-plane-protocols/rt:control-plane-protocol/"
    + "ospf:ospf/ospf:areas/"
    + "ospf:area/ospf:database/"
    + "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
    + "ospf:area-scope-lsa/ospf:version/ospf:ospfv2/"
    + "ospf:ospfv2/ospf:body/ospf:opaque/ospf:te-opaque/"
    + "ospf:link-tlv" {
when "../.../.../.../.../.../.../.../.../.../.../..."
    + "rt:type = 'ospf:ospfv2'" {
    description
        "This augmentation is only valid for OSPFv2.";
    }
description
    "Traffic Engineering Network-to-Router Sub-TLV.";

```

```

    leaf network-to-router-te-metric {
        type uint32;
        description "Network to Router TE metric.";
        reference
            "RFC 8042 - OSPF Two-Part Metric";
    }
}

augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:areas/"
+ "ospf:area/ospf:database/"
+ "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
+ "ospf:area-scope-lsa/ospf:version/ospf:ospfv3/"
+ "ospf:ospfv3/ospf:body/ospfv3-te:ospfv3-intra-area-te/"
+ "ospfv3-te:link-tlv" {
when "../.../.../.../.../.../.../.../.../.../..."
+ "rt:type = 'ospf:ospfv3'" {
    description
        "This augmentation is only valid for OSPFv3.";
}
description
    "Traffic Engineering Network-to-Router Sub-TLV.";
    leaf network-to-router-te-metric {
        type uint32;
        description "Network to Router TE metric.";
        reference
            "RFC 8042 - OSPF Two-Part Metric";
    }
}
}
}
<CODE ENDS>

```

4. YANG Module for OSPF Graceful Link Shutdown

This document defines a YANG module for OSPF Graceful Link Shutdown feature as defined in [RFC8379]. It is an augmentation of the OSPF base model.

```

module: ietf-ospf-graceful-link-shutdown
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
    /ospf:interfaces/ospf:interface:
    +--rw graceful-link-shutdown
      +--rw enable?    boolean
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
    /ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
    /ospf:area-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
    /ospf:body/ospf:opaque/ospf:extended-link-opaque
    /ospf:extended-link-tlv:
    +--ro graceful-link-shutdown-sub-tlv!
    +--ro remote-address-sub-tlv
      | +--ro remote-address?    inet:ipv4-address
    +--ro local-remote-int-id-sub-tlv
      +--ro local-int-id?      uint32
      +--ro remote-int-id?    uint32
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
    /ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
    /ospf:area-scope-lsa/ospf:version/ospf:ospfv3/ospf:ospfv3
    /ospf:body/ospfv3-e-lsa:e-router/ospfv3-e-lsa:e-router-tlvs
    /ospfv3-e-lsa:link-tlv:
    +--ro graceful-link-shutdown-sub-tlv!
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/ospf:ospf/ospf:database
    /ospf:as-scope-lsa-type/ospf:as-scope-lsas/ospf:as-scope-lsa
    /ospf:version/ospf:ospfv3/ospf:ospfv3/ospf:body
    /ospfv3-e-lsa:e-router/ospfv3-e-lsa:e-router-tlvs
    /ospfv3-e-lsa:link-tlv:
    +--ro graceful-link-shutdown-sub-tlv!

```

<CODE BEGINS> file "ietf-ospf-graceful-link-shutdown@2019-08-13.yang"

```

module ietf-ospf-graceful-link-shutdown {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-ospf-graceful-link-shutdown";

  prefix ospf-grace-linkdown;

  import ietf-inet-types {
    prefix "inet";
  }

  import ietf-routing {
    prefix "rt";
  }

```

```
}

import ietf-ospf {
  prefix "ospf";
}

import ietf-ospfv3-extended-lsa {
  prefix "ospfv3-e-lsa";
}

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

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

   Author:    Yingzhen Qu
               <mailto:yqu@futurewei.com>
   Author:    Acee Lindem
               <mailto:acee@cisco.com>";

description
  "This YANG module defines the configuration and operational
   state for OSPF Graceful Link Shutdown feature as defined
   in RFC 8379."

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

```

}

/* RFC 8379 */
augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/ospf:ospf/"
  + "ospf:areas/ospf:area/ospf:interfaces/ospf:interface" {
  when "../.../rt:type = 'ospf:ospfv2' or "
    + "../.../rt:type = 'ospf:ospfv3'" {
    description
      "This augments the OSPF interface configuration
      when used.";
  }
  description
    "This augments the OSPF protocol interface
    configuration with segment routing.";

  container graceful-link-shutdown {
    leaf enable {
      type boolean;
      default false;
      description
        "Enable OSPF graceful link shutdown.";
    }
    description
      "OSPF Graceful Link Shutdown.";
  }
}

/* Database */
augment "/rt:routing/"
  + "rt:control-plane-protocols/rt:control-plane-protocol/"
  + "ospf:ospf/ospf:areas/"
  + "ospf:area/ospf:database/"
  + "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
  + "ospf:area-scope-lsa/ospf:version/ospf:ospfv2/"
  + "ospf:ospfv2/ospf:body/ospf:opaque/"
  + "ospf:extended-link-opaque/ospf:extended-link-tlv" {
  when "../.../rt:type = 'ospf:ospfv2'" {
    description
      "This augmentation is only valid for OSPFv2.";
  }
  description
    "OSPF graceful link shutdown for OSPFv2 extended link TLV
    in type 10 opaque LSA.";
}

```

```
    container graceful-link-shutdown-sub-tlv {
      presence "Enable graceful link shutdown";
      description
        "Graceful-Link-Shutdown sub-TLV identifies the link as being
        gracefully shutdown.";
    }

    container remote-address-sub-tlv {
      leaf remote-address {
        type inet:ipv4-address;
        description
          "Remote IPv4 address used to identify a particular link
          on the remote side.";
      }
      description
        "This sub-TLV specifies the IPv4 address of the remote
        endpoint on the link.";
    }

    container local-remote-int-id-sub-tlv {
      leaf local-int-id {
        type uint32;
        description "Local interface ID.";
      }
      leaf remote-int-id {
        type uint32;
        description "Remote interface ID.";
      }
      description
        "This sub-TLV specifies Local and Remote Interface IDs.";
    }
  }

  augment "/rt:routing/"
    + "rt:control-plane-protocols/rt:control-plane-protocol/"
    + "ospf:ospf:areas/ospf:area/ospf:database/"
    + "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
    + "ospf:area-scope-lsa/ospf:version/ospf:ospfv3/"
    + "ospf:ospfv3/ospf:body/ospfv3-e-lsa:e-router"
    + "/ospfv3-e-lsa:e-router-tlvs/ospfv3-e-lsa:link-tlv" {
    when "'ospf:.../.../.../.../.../.../.../.../.../...'"
      + "rt:type" = 'ospf:ospfv3' {
      description
        "This augmentation is only valid for OSPFv3
        E-Router LSAs";
    }
  }

  container graceful-link-shutdown-sub-tlv {
    presence "Enable graceful link shutdown";
```



```

        description
            "Graceful-Link-Shutdown sub-TLV identifies the link as being
            gracefully shutdown.";
    }
    description
        "Augemnt OSPFv3 Area scope router-link TLV.";
}

augment "/rt:routing/"
    + "rt:control-plane-protocols/rt:control-plane-protocol/"
    + "ospf:ospf/ospf:database/"
    + "ospf:as-scope-lsa-type/ospf:as-scope-lsas/"
    + "ospf:as-scope-lsa/ospf:version/ospf:ospfv3/"
    + "ospf:ospfv3/ospf:body/ospfv3-e-lsa:e-router"
    + "/ospfv3-e-lsa:e-router-tlvs/ospfv3-e-lsa:link-tlv" {
    when "'ospf:.../.../.../.../.../.../.../.../"
        + "rt:type" = "ospf:ospfv3" {
        description
            "This augmentation is only valid for OSPFv3
            E-Router LSAs";
    }
}

container graceful-link-shutdown-sub-tlv {
    presence "Enable graceful link shutdown";
    description
        "Graceful-Link-Shutdown sub-TLV identifies the link as being
        gracefully shutdown.";
}

description
    "Augemnt OSPFv3 AS scope router-link TLV.";
}
}

<CODE ENDS>

```

5. YANG Module for OSPF LLS Extension for Local Interface ID Advertisement

This document defines a YANG module for OSPF Link-Local Signaling (LLS) Extensions for Local Interface ID Advertisement feature as defined in [RFC8510]. It is an augmentation of the OSPF base model.

```
module: ietf-ospf-lls-local-id
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/ospf:ospf:
      +--rw lls-int-id
        +--rw enable?    boolean
```

```
<CODE BEGINS> file "ietf-ospf-lls-local-id@2019-08-13.yang"
```

```
module ietf-ospf-lls-local-id {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-ospf-lls-local-id";

  prefix ospf-lls-localid;

  import ietf-routing {
    prefix "rt";
  }

  import ietf-ospf {
    prefix "ospf";
  }

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

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

    Author:   Yingzhen Qu
              <mailto:yqu@futurewei.com>
    Author:   Acee Lindem
              <mailto:acee@cisco.com>";

  description
    "This YANG module defines the configuration and operational
    state for OSPF Link-Local Signaling (LLS) Extensions for Local
    Interface ID Advertisement feature as defined in RFC 8510.

    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 version";
  reference
    "RFC XXXX: A YANG Data Model for OSPF.";
}

augment "/rt:routing/rt:control-plane-protocols"
  + "/rt:control-plane-protocol/ospf:ospf" {
  when "../rt:type = 'ospf:ospfv2' or "
    + "../rt:type = 'ospf:ospfv3'" {
    description
      "This augments the OSPF routing protocol when used.";
  }
  description
    "This augments the OSPF protocol configuration
    to support LLS extensions for interface ID as
    defined in RFC 8510.";
  container lls-int-id {
    leaf enable {
      type boolean;
      default false;
      description
        "Enable LLS to advertise local interface ID.";
    }
    description
      "OSPF LLS Extensions for interface ID.";
  }
}
}
}
<CODE ENDS>

```

6. Security Considerations

The YANG modules specified in this document define a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC5246].

The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF or RESTCONF users to a pre-configured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in the modules that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable

in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations.

Some of the readable data nodes in the modules may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. The exposure of the Link State Database (LSDB) will expose the detailed topology of the network. This may be undesirable since both due to the fact that exposure may facilitate other attacks. Additionally, network operators may consider their topologies to be sensitive confidential data.

7. IANA Considerations

This document registers URIs in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registrations is requested to be made:

URI: urn:ietf:params:xml:ns:yang:ietf-ospf-two-metric
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-ospf-grace-linkdown
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-ospf-lls-localid
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

This document registers the YANG modules in the YANG Module Names registry [RFC6020].

name: ietf-ospf-two-metric
namespace: urn:ietf:params:xml:ns:yang:ietf-ospf-two-metric
prefix: ospf-two-metric
reference: RFC XXXX

name: ietf-ospf-grace-linkdown
namespace: urn:ietf:params:xml:ns:yang:ietf-ospf-grace-linkdown
prefix: ospf-grace-linkdown
reference: RFC XXXX

name: ietf-ospf-lls-localid
namespace: urn:ietf:params:xml:ns:yang:ietf-ospf-lls-localid
prefix: ospf-lls-localid
reference: RFC XXXX

8. 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.

9. References

9.1. 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>>.
- [RFC5329] Ishiguro, K., Manral, V., Davey, A., and A. Lindem, Ed., "Traffic Engineering Extensions to OSPF Version 3", RFC 5329, DOI 10.17487/RFC5329, September 2008, <<https://www.rfc-editor.org/info/rfc5329>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <<https://www.rfc-editor.org/info/rfc6536>>.

- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [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>>.
- [RFC8042] Zhang, Z., Wang, L., and A. Lindem, "OSPF Two-Part Metric", RFC 8042, DOI 10.17487/RFC8042, December 2016, <<https://www.rfc-editor.org/info/rfc8042>>.
- [RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <<https://www.rfc-editor.org/info/rfc8349>>.
- [RFC8379] Hegde, S., Sarkar, P., Gredler, H., Nanduri, M., and L. Jalil, "OSPF Graceful Link Shutdown", RFC 8379, DOI 10.17487/RFC8379, May 2018, <<https://www.rfc-editor.org/info/rfc8379>>.
- [RFC8510] Psenak, P., Ed., Talaulikar, K., Henderickx, W., and P. Pillay-Esnault, "OSPF Link-Local Signaling (LLS) Extensions for Local Interface ID Advertisement", RFC 8510, DOI 10.17487/RFC8510, January 2019, <<https://www.rfc-editor.org/info/rfc8510>>.

9.2. 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

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

EMail: yingzhen.qu@futurewei.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

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

EMail: 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

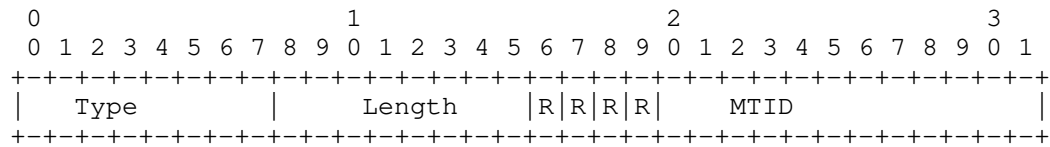
6. Advertising Locators and End SIDs

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

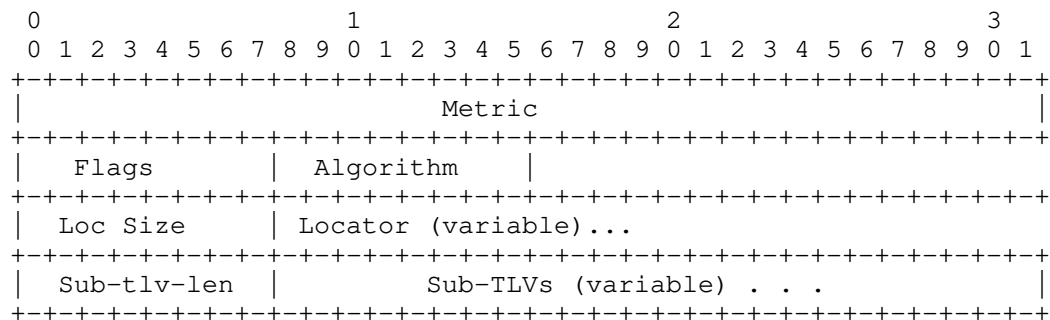
This new TLV shares the sub-TLV space defined for TLVs 135, 235, 236 and 237.

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)

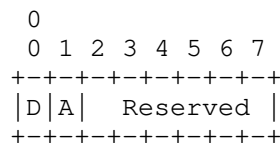
Length: variable.

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

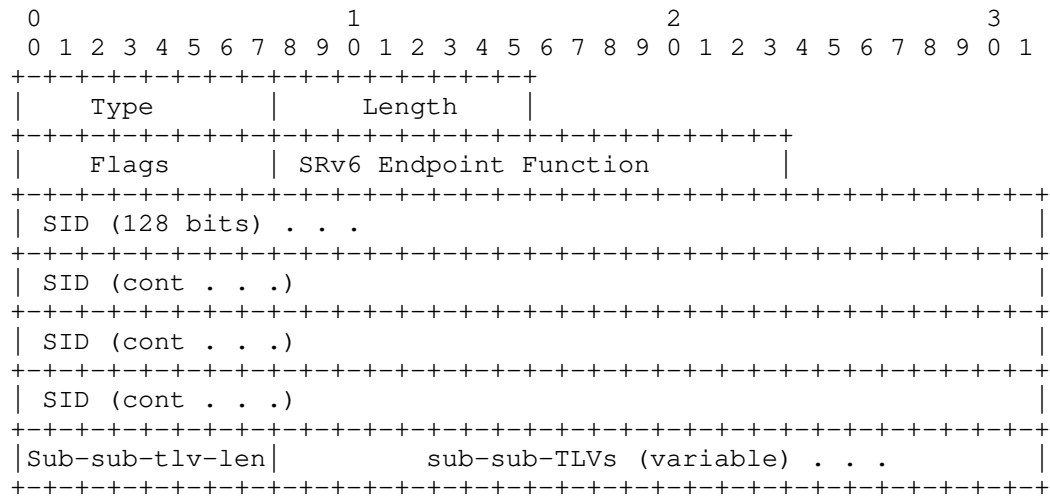
Optional sub-TLVs.

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	2	3	4	5	6	7	8	9																								
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.

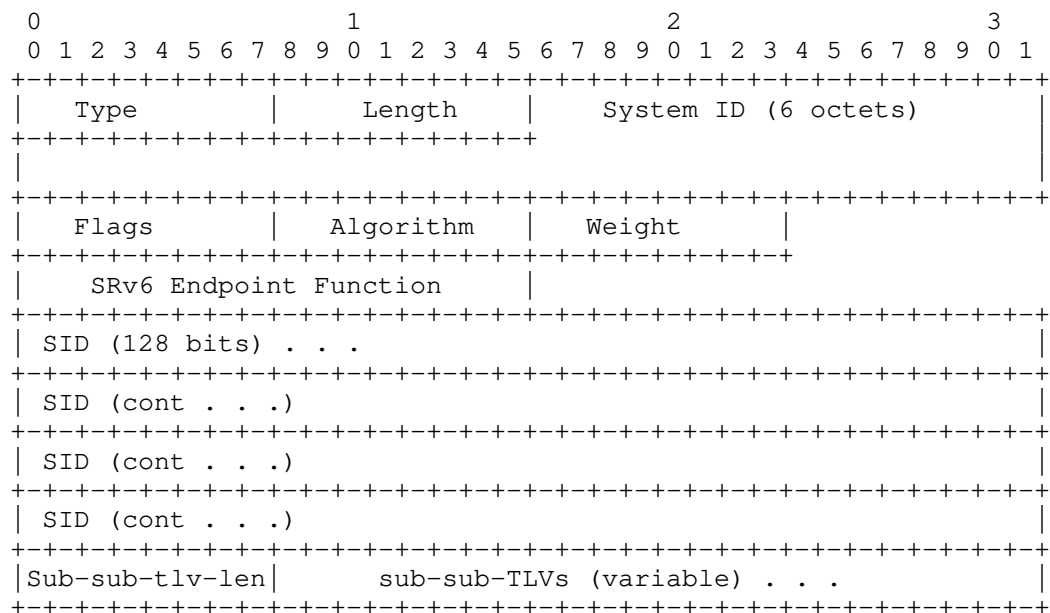
Sub-sub-TLV-length: 1 octet. Number of octets used by sub-sub-TLVs

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:

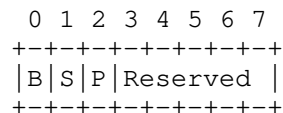


Type: 44 (Suggested value to be assigned by IANA)

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 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,
<<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

LSR Working Group
Internet-Draft
Intended status: Standards Track
Expires: 28 August 2022

P. Kaneriya
R. Shetty
S. Hegde
R. Bonica
Juniper Networks
24 February 2022

IS-IS Extensions To Support The IPv6 Compressed Routing Header (CRH)
draft-bonica-lsr-crh-isis-extensions-06

Abstract

Source nodes can use the IPv6 Compressed Routing Header (CRH) to steer packets through a specified path. This document defines IS-IS extensions that support the CRH.

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 28 August 2022.

Copyright Notice

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

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

Table of Contents

1. Introduction	2
2. Requirements Language	2
3. Advertising The CRH Capability	2
4. Advertising Prefix Segment Identifiers	3
5. Advertising Adjacency Segment Identifiers	4
6. Advertising Adjacency Segment Identifiers Into LANs	5
7. IANA Considerations	7
7.1. The CRH Sub-TLV	7
7.2. Prefix SID Sub-TLV	8
7.3. Adjacency SID Sub-TLV	8
8. Security Considerations	9
9. Acknowledgements	9
10. References	9
10.1. Normative References	9
10.2. Informative References	11
Authors' Addresses	11

1. Introduction

Source nodes can use the IPv6 Compressed Routing Header (CRH) [I-D.bonica-6man-comp-rtg-hdr] to steer packets through a specified path. This document defines IS-IS extensions that support the CRH.

2. Requirements Language

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

3. Advertising The CRH Capability

The Router CAPABILITY TLV [RFC7981] MAY contain exactly one CRH sub-TLV. The CRH sub-TLV indicates that the advertising node can process the CRH.

The CRH sub-TLV MAY contain sub-sub-TLVs. No sub-sub-TLVs are currently defined.

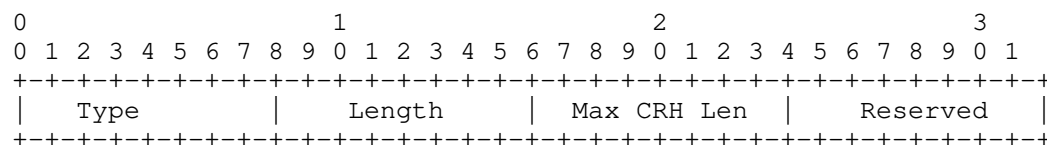


Figure 1: CRH Sub-TLV

Figure 1 depicts the CRH sub-TLV. The CRH sub-TLV contains the following fields:

- * Type: 8 bits. CRH (value TBD by IANA. Suggested value is 30.)
- * Length: 8 bits. Length of TLV data excluding the TLV header. MUST be equal to 2 plus the length of sub-sub-TLVs (if any).
- * Max CRH Len: 8 bits. Maximum CRH length supported by the advertising node, measured in 8-octet units, not including the first 8 octets. See Note 1.
- * Reserved: 8 bits. SHOULD be set to zero by sender. MUST be ignored by receiver.

Note 1: According to [RFC8200], all IPv6 Routings header include a "Hdr Ext Len" field. That field specifies the length of the Routing header in 8-octet units, not including the first 8 octets. The same unit of measure was chosen for the "Max CRH Len" field in the CRH sub-TLV.

4. Advertising Prefix Segment Identifiers

The following TLVs MAY contain one or more Prefix SID sub-TLVs:

- * TLV-236 (IPv6 IP Reachability) [RFC5308].
- * TLV-237 (Multitopology IPv6 IP Reachability) [RFC5120].

The Prefix SID sub-TLV is valid only when its parent TLV specifies a prefix length of 128. In this case, it binds the SID that it contains to the prefix (i.e., IPv6 address) that its parent TLV contains. This information is used to construct the mapping table described in [I-D.bonica-6man-comp-rtg-hdr].

When the parent TLV is propagated across level boundaries, the Prefix SID sub-TLV SHOULD be kept.

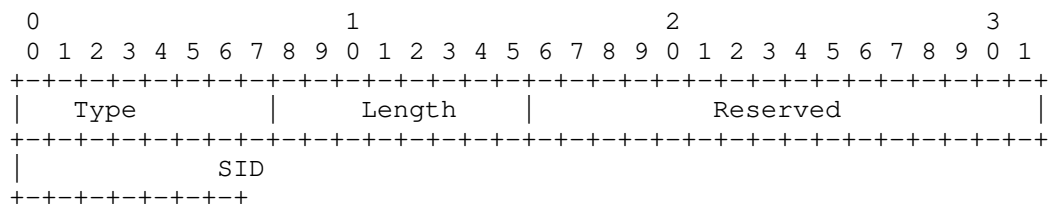


Figure 2: Prefix SID sub-TLV

Figure 2 depicts the Prefix SID sub-TLV. It contains the following fields:

- * Type: 8 bits. Prefix SID sub-TLV (Value TBD by IANA. Suggested value is 33.)
- * Length: 8 bits. Length of TLV data excluding the TLV header, measured in bytes.
- * Reserved: 16 bits. SHOULD be set to zero by the sender. MUST be ignored by the receiver.
- * SID - Variable length. Segment Identifier.

5. Advertising Adjacency Segment Identifiers

The following TLVs can contain one or more Adjacency SID sub-TLVs:

- * TLV-22 (Extended IS reachability) [RFC5305]
- * TLV-222 (Multitopology IS) [RFC5120]
- * TLV-23 (IS Neighbor Attribute) [RFC5311]
- * TLV-223 (Multitopology IS Neighbor Attribute) [RFC5311]
- * TLV-141 (inter-AS reachability information) [RFC5316]

The Adjacency SID sub-TLV is valid only when its parent TLV also contains an IPv6 Neighbor Address sub-TLVs [RFC6119]. In this case, the SID contained by the Adjacency SID sub-TLV is bound to the IPv6 address contained by the IPv6 Neighbor Address sub-TLV. This information is used to construct the mapping table described in [I-D.bonica-6man-comp-rtg-hdr].

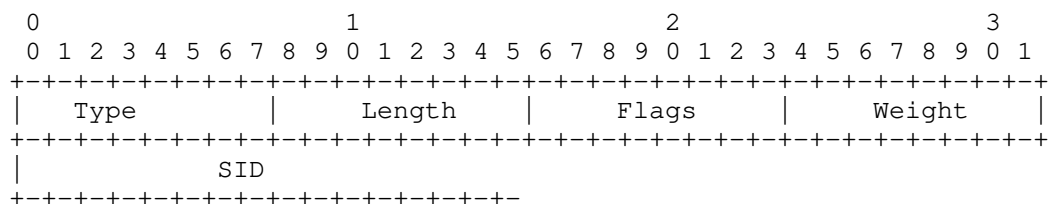


Figure 3: Adjacency SID Sub-TLV

Figure 3 depicts the Adjacency SID sub-TLV. It contains the following fields:

- * Type: 8 bits. Adjacency SID sub-TLV (Value TBD by IANA. Suggested value is 45.)
- * Length: 8 bits. Length of TLV data excluding the TLV header, measured in bytes.
- * Flags: 8 bits. See below.
- * Weight: 8 bits. The value represents the SID weight for the purpose of load balancing.
- * SID - Variable length. Segment Identifier.

```

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

```

Figure 4: Adjacency SID Sub-TLV Flags

Figure 4 depicts Adjacency SID Sub-TLV flags. They include the following:

- * B-Flag: Backup flag. If set, the SID is eligible for protection.
- * S-Flag: Set flag. When set, the S-Flag indicates that the SID refers to a set of adjacencies (and therefore MAY be assigned to other adjacencies as well).
- * P-Flag: Persistent flag. When set, the P-Flag indicates that the SID is persistently allocated, i.e., the SID value remains consistent across router restart and/or interface flap.)

6. Advertising Adjacency Segment Identifiers Into LANs

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

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

The following TLVs can contain one or more LAN Adjacency SID sub-TLVs:

- * TLV-22 (Extended IS reachability) [RFC5305]
- * TLV-222 (Multitopology IS) [RFC5120]
- * TLV-23 (IS Neighbor Attribute) [RFC5311]
- * TLV-223 (Multitopology IS Neighbor Attribute) [RFC5311]

The LAN Adjacency SID sub-TLV binds an IPv6 address to a SID. The sub-TLV contains both the IPv6 address and the SID. This information is used to construct the mapping table described in [I-D.bonica-6man-comp-rtg-hdr].

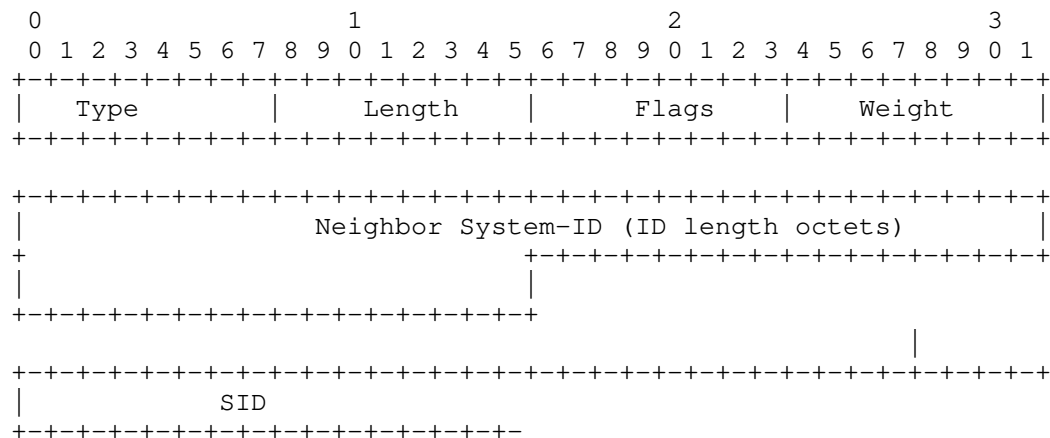


Figure 5: LAN Strictly Routed SID Sub-TLV

Figure 5 depicts the Adjacency SID sub-TLV. It contains the following fields:

- * Type: 8 bits. Adjacency SID sub-TLV (Value TBD by IANA. Suggested value is 46.)
- * Length: 8 bits. Length of TLV data excluding the TLV header, measured in bytes.
- * Flags: 8 bits. See below.
- * Weight: 8 bits. The value represents the SID weight for the purpose of load balancing.

- * Neighbor System-ID: 6 bytes. IS-IS System-ID of length "ID Length" as defined in [ISO10589].
- * SID - Variable length. Segment Identifier.

```

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

```

Figure 6: Adjacency SID Sub-TLV Flags

Figure 6 depicts Adjacency SID Sub-TLV flags. They include the following:

- * B-Flag: Backup flag. If set, the SID is eligible for protection.
- * S-Flag: Set flag. When set, the S-Flag indicates that the SID refers to a set of adjacencies (and therefore MAY be assigned to other adjacencies as well).
- * P-Flag: Persistent flag. When set, the P-Flag indicates that the SID is persistently allocated, i.e., the SID value remains consistent across router restart and/or interface flap.)

7. IANA Considerations

7.1. The CRH Sub-TLV

IANA is requested to add a new sub-TLV in the Sub-TLVs for TLV 242 (IS-IS Router CAPABILITY TLV) Registry [capreg].

- * Value - TBD by IANA. (Suggested value is 30).
- * Description - CRH

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

- * 0 - Reserved
- * 1 - 255 Unassigned

7.2. Prefix SID Sub-TLV

IANA is requested to add a new entry in the Sub-TLVs for TLVs 135, 235, 236, and 237 (Extended IP reachability, MT IP. Reach, IPv6 IP. Reach, and MT IPv6 IP. Reach TLVs) Registry [loosereg].

- * Value - TBD by IANA. (Suggested value is 33)
- * Description - Prefix SID
- * 135 - N
- * 136 - N
- * 236 - Y
- * 237 - Y
- * Reference - This document.

7.3. Adjacency SID Sub-TLV

IANA is requested to add the following entries in the Sub-TLVs for TLVs 22, 23, 25, 141, 222, and 223 (Extended IS reachability, IS Neighbor Attribute, L2 Bundle Member Attributes, inter-AS reachability information, MT-ISN, and MT IS Neighbor Attribute TLVs) Registry [strictreg].

The first entry follows:

- * Value - TBD by IANA (Suggested value is 45).
- * Description - Adjacency SID
- * 22 - Y
- * 23 - Y
- * 25 - N
- * 141 - Y
- * 222 - Y
- * 223 - Y
- * Reference - This document.

The second entry follows:

- * Value - TBD by IANA (Suggested value is 46)
- * Description - LAN Adjacency SID
- * 22 - Y
- * 23 - Y
- * 25 - N
- * 141 - N
- * 222 - Y
- * 223 - Y
- * Reference - This document.

8. Security Considerations

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

9. Acknowledgements

Thanks to Ram Santhanakrishnan for his comments on this document.

10. References

10.1. Normative References

[I-D.bonica-6man-comp-rtg-hdr]

Bonica, R., Kamite, Y., Alston, A., Henriques, D., and L. Jalil, "The IPv6 Compact Routing Header (CRH)", Work in Progress, Internet-Draft, draft-bonica-6man-comp-rtg-hdr-27, 15 November 2021, <<https://www.ietf.org/archive/id/draft-bonica-6man-comp-rtg-hdr-27.txt>>.

[ISO10589] IANA, "Intermediate system to Intermediate system routing information exchange protocol for use in conjunction with the Protocol for providing the Connectionless-mode Network Service (ISO 8473)", August 1987, <ISO/IEC 10589:2002>.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)", RFC 5120, DOI 10.17487/RFC5120, February 2008, <<https://www.rfc-editor.org/info/rfc5120>>.
- [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>>.
- [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>>.
- [RFC5311] McPherson, D., Ed., Ginsberg, L., Previdi, S., and M. Shand, "Simplified Extension of Link State PDU (LSP) Space for IS-IS", RFC 5311, DOI 10.17487/RFC5311, February 2009, <<https://www.rfc-editor.org/info/rfc5311>>.
- [RFC5316] Chen, M., Zhang, R., and X. Duan, "ISIS Extensions in Support of Inter-Autonomous System (AS) MPLS and GMPLS Traffic Engineering", RFC 5316, DOI 10.17487/RFC5316, December 2008, <<https://www.rfc-editor.org/info/rfc5316>>.
- [RFC6119] Harrison, J., Berger, J., and M. Bartlett, "IPv6 Traffic Engineering in IS-IS", RFC 6119, DOI 10.17487/RFC6119, February 2011, <<https://www.rfc-editor.org/info/rfc6119>>.
- [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>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.

10.2. Informative References

- [capreg] IANA, "Sub-TLVs for TLV 242 (IS-IS Router CAPABILITY TLV)", August 1987, <<https://www.iana.org/assignments/isis-tlv-codepoints/isis-tlv-codepoints.xhtml#isis-tlv-codepoints-242>>.
- [loosereg] IANA, "Sub-TLVs for TLVs 135, 235, 236, and 237 (Extended IP reachability, MT IP. Reach, IPv6 IP. Reach, and MT IPv6 IP. Reach TLVs)", August 1987, <<https://www.iana.org/assignments/isis-tlv-codepoints/isis-tlv-codepoints.xhtml#isis-tlv-codepoints-135-235-236-237>>.
- [strictreg] IANA, "Sub-TLVs for TLVs 22, 23, 25, 141, 222, and 223 (Extended IS reachability, IS Neighbor Attribute, L2 Bundle Member Attributes, inter-AS reachability information, MT-ISN, and MT IS Neighbor Attribute TLVs)", August 1987, <<https://www.iana.org/assignments/isis-tlv-codepoints/isis-tlv-codepoints.xhtml#isis-tlv-codepoints-22-23-25-141-222-223>>.

Authors' Addresses

Parag Kaneriya
Juniper Networks
Elnath-Exora Business Park Survey
Bangalore 560103
Karnataka
India
Email: pkaneria@juniper.net

Rejesh Shetty
Juniper Networks
Elnath-Exora Business Park Survey
Bangalore 560103
Karnataka
India
Email: mrajesh@juniper.net

Shraddha Hegde
Juniper Networks
Elnath-Exora Business Park Survey
Bangalore 560103
Karnataka
India
Email: shraddha@juniper.net

Ron Bonica
Juniper Networks
2251 Corporate Park Drive
Herndon, Virginia 20171
United States of America
Email: rbonica@juniper.net

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

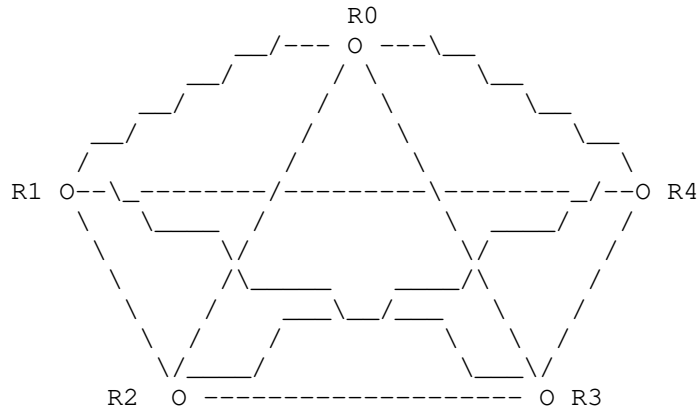
- [I-D.ietf-rtgwg-spf-uloop-pb-statement]
Litkowski, S., Decraene, B., and M. Horneffer, "Link State protocols SPF trigger and delay algorithm impact on IGP micro-loops", draft-ietf-rtgwg-spf-uloop-pb-statement-10 (work in progress), January 2019.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines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

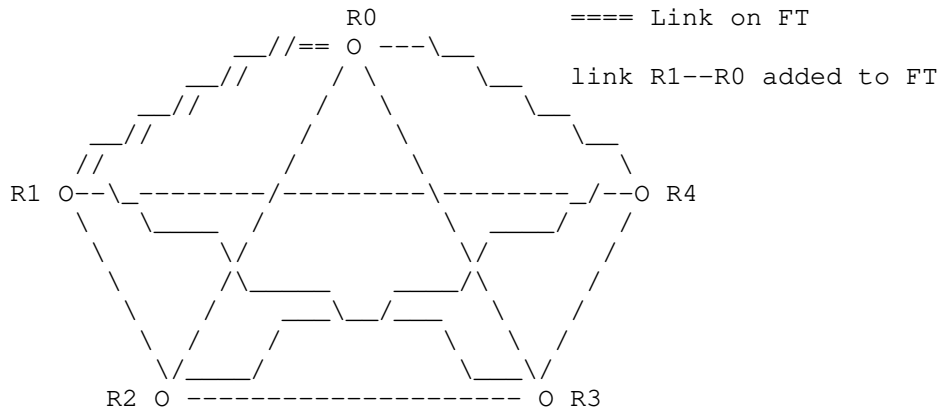
ID < R1's ID < R2's ID < R3's ID < R4's ID. The algorithm starts with $Cq = \{(R0, D=0, PH=\{\})\}$, $FT = \{\}$, $MaxD = 3$.

```
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
         ^^^          ^^^^^^^^^^^^^^^

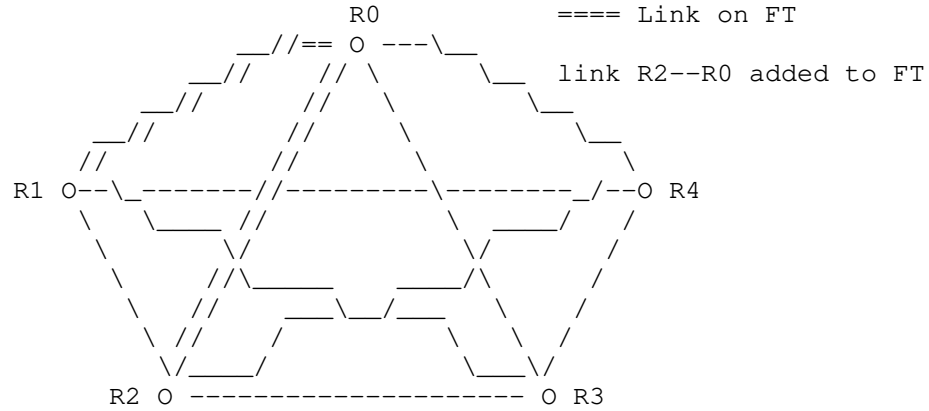
   // for Ri connected to R1 (in Cq) not on FT, append R1 to Ri's PHs
   Cq = { (R2,0, {R0,R1}), (R3,0, {R0,R1}), (R4,0,{R0,R1}) }.
         ^^          ^^          ^^
```



```

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}) }
           ^^               ^^

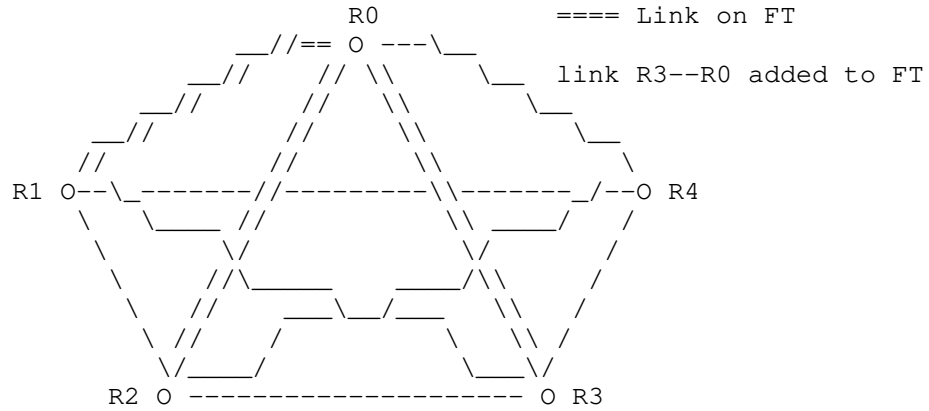
```



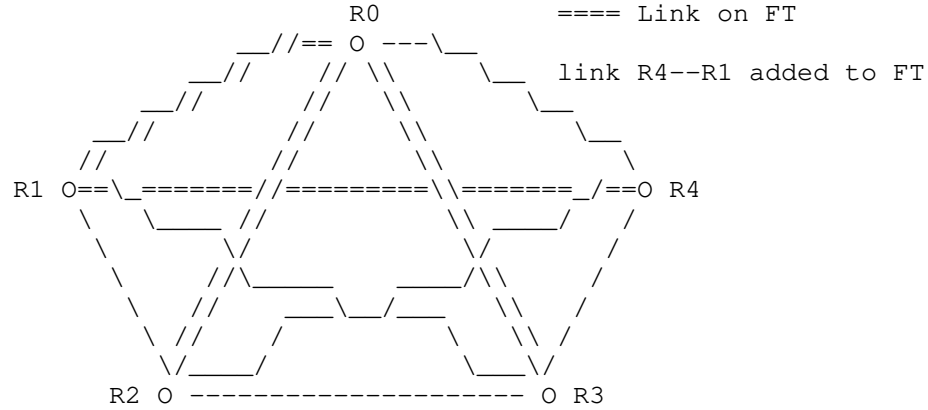
```

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}) }.
           ^^

```

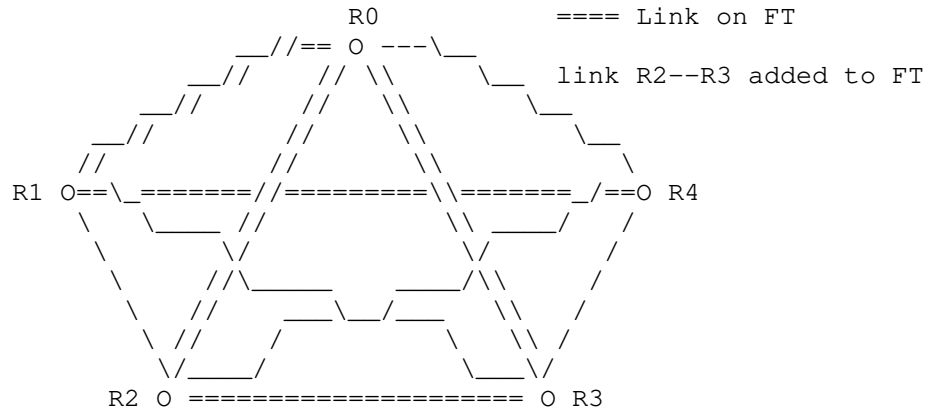


4. //remove the 1st element (R4,0,{R0,R1,R2,R3}) from Cq,R1's D=1 < MaxD
 Cq = { }
 // add (R4,1,{R1}) into FT, increase R1's D by one
 FT = {(R0,3,{})}, (R1,2,{R0}), (R2,1,{R0}), (R3,1,{R0}), (R4,1,{R1})}



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.

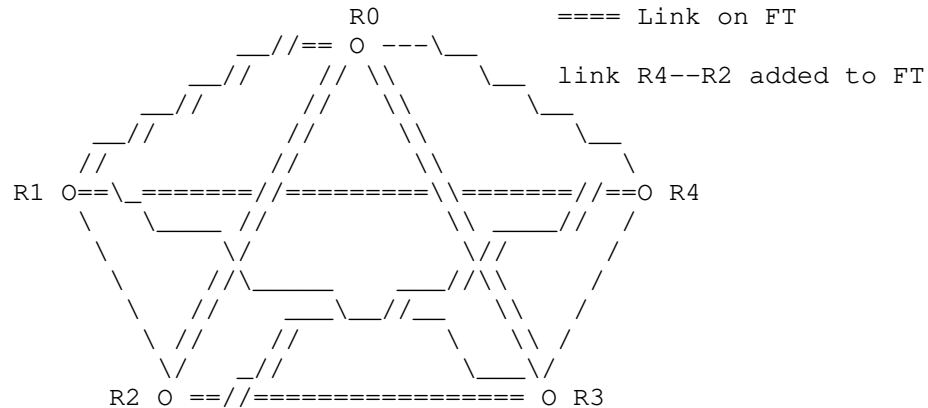
5. // On FT, get node R2 with smallest ID whose D=1
 FT = {(R0,3,{})}, (R1,2,{R0}), (R2,1,{R0}), (R3,1,{R0}), (R4,1,{R1})}
 // Add link R2--R3 to FT, ^^^^^^^^^^^
 // where R2--R3 is not on FT, R3's D=1 is minimum first and then
 // R3's ID is minimum (R3 and R4 tie for D), R2's D++ and R3's D++
 FT =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

Mehmet Toy
Verizon
USA

Email: mehmet.toy@verizon.com

Yi Yang
IBM
Cary, NC
United States of America

Email: 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

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 13, 2022

B. Decraene
Orange
C. Bowers
Jayesh. J
Juniper Networks, Inc.
T. Li
Arista Networks
G. Van de Velde
Nokia
G. Solignac
Orange
July 12, 2021

IS-IS Flooding Congestion Control
draft-decraene-lsr-isis-flooding-speed-07

Abstract

This document proposes a mechanism to adjust IS-IS flooding speed between two adjacent routers by adjusting the sender flooding speed to the capability of the receiver. This helps improving the flooding throughput, reducing LSPs losses and retransmissions due to receiver overload, and avoiding manual tuning of flooding parameters by the network operator. This document defines a new TLV for Hello messages. This TLV may carry a set of parameters indicating the performance capacity to receive LSPs.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 13, 2022.

Copyright Notice

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

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

Table of Contents

1. Introduction	3
1.1. Requirements Language	4
2. Overview	4
3. Flooding Parameters TLV	5
3.1. InterfaceLSPReceiveWindow sub-TLV	5
3.2. minimumInterfaceLSPTransmissionInterval sub-TLV	5
4. Flow control	6
4.1. Operation on a point to point interface	6
4.2. Faster acknowledgments of LSPs	7
4.3. Operation on a LAN interface	8
5. Congestion control	9
5.1. Slow start	9
5.2. Congestion avoidance	10
5.3. Remarks	11
6. Interaction with other LSP rate limiting mechanisms	11
7. Determining values to be advertised in the Flooding Parameters TLV	12
8. Operation considerations	13
9. IANA Considerations	13
10. Security Considerations	13
11. Acknowledgments	14
12. References	15
12.1. Normative References	15
12.2. Informative References	15
Appendix A. Changes / Author Notes	16
Authors' Addresses	17

1. Introduction

IGP flooding is paramount for Link State IGP as routing computations assume that the Link State DataBases (LSDBs) are always in sync across all nodes in the flooding domain.

Slow flooding directly translates to delayed network reaction to failure and LSDB inconsistencies across nodes. The former increases packet loss. The latter translates to routing inconsistencies and possibly micro-loops leading to packet loss, link overload, and jitter for all classes of service. Note that across the network, multiple links may be affected by these forwarding issues, even in the case of a single link failure.

In addition, one single event in the network can require the flooding of multiple LSPs. The typical case is a node failure which requires the flooding of at least one LSP per neighbor of the failed node. Hence, if a node has N IGP neighbors, the failure of this node requires the advertisement and flooding of at least N LSPs. The network won't be able to converge to the new topology until all N LSPs are received by all nodes. Hence there is a need to be able to quickly exchange N LSPs. This document addresses this requirement by allowing the fast flooding of a number of consecutive LSPs.

IGP flooding is hard. One would want fast flooding when the network is stable and slow enough flooding to not overload the neighbor(s) when the network is less stable. Since flooding is performed hop by hop, not overloading the adjacent receiver is sufficient.

Improving the communication speed and efficiency between IS-IS neighbors improves IS-IS scaling. These extensions do not compete with proposed extensions to reduce LSP flooding traffic by reducing the flooding topology such as [I-D.ietf-lsr-dynamic-flooding]. On the contrary, this extension complements those proposals. Indeed reducing the flooding topology does not reduce the size of the LSDB or the total number of LSPs to exchange between two nodes. So increasing the overall flooding speed can be beneficial for nodes implementing dynamic flooding. The reverse is also true: as dynamic flooding reduces the number of neighbors with flooding enabled, this allows nodes implementing the flooding parameter extensions to focus their flooding resources on those neighbors by sending better parameters to the selected flooding nodes and worse parameters to non-selected flooding nodes.

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 RFC 2119 [RFC2119] RFC 8174 [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Overview

Ensuring the goodput between two entities is a layer 4 responsibility as per the OSI model and a typical example is the TCP protocol defined in RFC 793 [RFC0793] . It typically relies on the following sub-functions: flow control, congestion control and reliability.

Flow control is about creating a control loop between a single transmitter and single receiver. TCP provides a mean for the receiver to govern the amount of data sent by the sender. This is achieved by advertising a "receive window", in units of octets, with every ACK. This document proposes to use the same mechanism by advertising a receive window, in units of LSP packets, in IS-IS Hello. The window indicates an allowed number of LSPs that the sender may transmit before receiving acknowledgment of those LSPs. There is an assumption that this is related to the currently available data buffer space available for this adjacency. Indicating a large window encourages transmissions.

Congestion control is about creating multiple interacting control loops between multiple transmitters and multiple receivers. Whereas flow control prevents the sender from overwhelming the receiver, congestion control prevents senders from overwhelming the network. For an IS-IS adjacency, the network between two IS-IS neighbors is relatively limited in scope and consist in a link which is typically over-sized compared to the capability of the IS-IS speakers, but also includes components inside both routers such as a fabric switch, line card CPU and forwarding plane buffers which may experience congestion. This document proposes to use the AIMD (Additive Increase Multiplicative Decrease) algorithm to react to packet loss. Note that TCP Reno relies on the same algorithm.

Reliability relies on loss detection and recovery. IS-IS already has mechanisms to ensure the reliable transmission of LSPs. This is not changed by this document.

3. Flooding Parameters TLV

This document defines a new TLV called "Flooding Parameters TLV" that may be included in IIH PDUs. It allows the LSP receiver to advertise receiver related parameters and capabilities which allows the LSP sender to better adapt to the receiver.

Type: TBD1.

Length: variable, the size in octet of the Value field.

Value: a list of sub-TLVs.

Two sub-TLVs are defined in this document.

3.1. InterfaceLSPReceiveWindow sub-TLV

The sub-TLV InterfaceLSPReceiveWindow advertises the maximum number of un-acknowledged LSPs that the node can receive/process with no separation interval between LSPs.

Type: 1.

Length: 4 octets.

Value: number of un-acknowledged LSPs which can be sent back to back.

Note that if an LSP has not been acknowledged and is sent again, it does not count twice. The reason is that this LSP is assumed to be lost and hence not in a buffer at the receiver.

3.2. minimumInterfaceLSPTransmissionInterval sub-TLV

The sub-TLV minimumInterfaceLSPTransmissionInterval advertises the minimum interval, in micro-seconds, between LSPs arrivals which can be processed/received on this interface, after the maximum number of un-acknowledged LSPs has been sent.

Type: 2.

Length: 4 octets.

Value: minimum interval, in micro-seconds, between two consecutive LSPs sent after the receive window has been used.

4. Flow control

Flow control is about creating a control loop between a single transmitter and single receiver. This document proposes to use a mechanism similar to the TCP receive window to allow the receiver to govern the amount of data sent by the sender. This receive window indicates an allowed number of LSPs that the sender may transmit before receiving acknowledgment of those LSPs. This receive window, in units of LSPs, is advertised in the sub-TLV InterfaceLSPReceiveWindow.

4.1. Operation on a point to point interface

By sending the InterfaceLSPReceiveWindow sub-TLV with a value N, the node advertises to its IS-IS neighbor, its ability to receive a maximum of N un-acknowledged LSPs from this neighbor, with no separation interval. This is akin to a reception window or sliding window in flow control. This value typically reflects the socket buffer size. Special care must be taken to let space for Hello and SNP PDUs if they share the same socket. In this case, this document suggests to advertise a Receive Window corresponding to half the size of the socket buffer.

By sending the minimumInterfaceLSPTransmissionInterval sub-TLV with a value T, the node advertises to its IS-IS neighbor, its ability to receive, after the receive window is full, LSPs separated by at least T micro-seconds from this neighbor.

The IS transmitter MUST NOT exceed these parameters. After having sent N un-acknowledged LSPs, it MUST send the following LSPs with an interval of at least T micro-seconds between each LSP.

Note however that if either the LSP transmitter or receiver does not adhere to these parameters, for example because of transient conditions, this causes no fatal condition to the operation of IS-IS. The worst case, the loss of LSP on the IS receiver, is already accounted for in [ISO10589]. As per [ISO10589], after a few seconds, respectively 2 and 10 by default in [ISO10589], neighbors will exchange PSNP (for point to point interface) or CSNP (for broadcast interface) and recover from the lost LSPs. This worst case (overrunning the receiver) should however be avoided as those additional seconds are impacting the network and the traffic as the LSDB is not fully synchronized. Hence it is better to err on the conservative side and to under-run the receiver rather than over-run it.

For a given IS-IS adjacency, the Flooding Parameters TLV does not need to be advertised in each IIH. The IS transmitter uses the

latest received value for each parameter (sub-TLV) until a new value is advertised by the IS receiver. Note however that IIH are not reliably exchanged, hence may never be received. For a parameter which has never been advertised, the IS transmitter use its local default value. That value SHOULD be configurable on a per node basis and MAY be configurable on a per interface basis.

4.2. Faster acknowledgments of LSPs

As per [ISO10589] , on point to point interfaces, the LSP receiver dynamically acknowledges the received LSPs by sending PSNP messages.

By acknowledging the LSPs before the InterfaceLSPReceiveWindow is exhausted, the receiver can achieve dynamic flow control and increase the flooding throughput without risking overloading any IS-IS router. If the InterfaceLSPReceiveWindow is large enough, the downstream flooding node can acknowledge a set of multiple LSPs up to the maximum size of a PSNP (90 LSPs) which allows dynamic flow control with limited or even no increase in the number of sent PSNPs.

In order to avoid reducing the throughput, the receiver should avoid letting the receive window exhaust. Therefore, the receiver SHOULD acknowledge the LSP more quickly than the default specified in [ISO10589] . This is beneficial both to the LSP sender which receives faster feedback and to the LSP receiver which has more time to acknowledge many LSPs before the sender times out and resend the LSP.

Receiver SHOULD reduce partialSNPInterval. The choice of this lower value is a local choice. It may depend on the (available) processing power of the node, the number of adjacencies, the requirement to synchronize the LSDB more quickly. 200 ms seems a reasonable value.

In addition to the timer based partialSNPInterval, the receiver SHOULD keep track of the number of unacknowledged LSPs per circuit and level. When this number exceeds a preset threshold LSP per PSNP (LPP), the receiver SHOULD immediately send a PSNP without waiting for the PSNP timer to expire. In case of a burst of LSPs, this allows for more frequent PSNPs, hence a faster feedback loop to the sender. In the absence of burst, the usual time-based PSNP approach comes into effect. This number SHOULD be lower than the advertised receive window InterfaceLSPReceiveWindow, e.g., InterfaceLSPReceiveWindow/2. This number SHOULD also be lower or equal to 90 as this is the maximum number of LSPs that can be acknowledged in a PSNP, hence waiting longer would not reduce the number of PSNPs sent but would delay the acknowledgements. Best performance is achieved when this number is an integer fraction of InterfaceLSPReceiveWindow. Based on experimental evidence, 15

unacknowledged LSPs is a right value assuming that InterfaceLSPReceiveWindow is at least twice bigger (>30).

By deploying both the time-based and the threshold-based PSNP approaches, the receiver can be adaptive to both LSP bursts and infrequent LSP updates.

4.3. Operation on a LAN interface

On a LAN interface, an IS receiver will generally receive LSPs from multiple IS transmitters. Also the LSPs sent by a given IS transmitter is received by all of the IS receivers on that LAN. In this section, we clarify how the flooding parameters should be interpreted in the context of a LAN.

An IS receiver on a LAN will communicate its desired flooding parameters using a single Flooding Parameters TLV, copies of which will be received by all N transmitters. The flooding parameters sent by the IS receiver MUST be understood as instructions from the receiver to each transmitter about the desired maximum transmit characteristics of each transmitter. The receiver is aware that there are N transmitters that can send LSPs to the receiver LAN interface. The receiver might want to take that into account by advertising a higher value of InterfaceLSPTransmissionInterval on this LAN interface than what it would advertise on a point to point interface. When the transmitters receive the InterfaceLSPTransmissionInterval value advertised by the DIS receiver, the transmitters should rate limit LSPs according to the advertised flooding parameters. They should not apply any further interpretation to the flooding parameters advertised by the receiver.

A given IS transmitter will receive flooding parameter advertisements from N different Flooding Parameters TLVs, which may carry different flooding parameter values. A given transmitter SHOULD use the most conservative value on a per Flooding parameter basis. For example, if the transmitter receives InterfaceLSPReceiveWindow from N IS-IS nodes on the LAN, it should use the smallest value.

In order for the InterfaceLSPReceiveWindow to be a useful parameter, an IS transmitter needs to be able to keep track of the number of unacknowledged LSPs it has sent to a given IS receiver. On a LAN there is no explicit acknowledgment of the receipt of LSPs between a given IS transmitter and a given IS receiver. However, an IS transmitter on a LAN can infer whether any IS receiver on the LAN has requested retransmission of LSPs from the DIS, by monitoring PSNPs generated on the LAN. If no PSNPs have been generated on the LAN for a suitable period of time, then an IS transmitter can safely set the number of unacknowledged LSPs to zero. Since this suitable period of time is

much higher than the fast acknowledgment of LSP defined in Section 4.2, the sustainable sending rate of LSP will be much slower on a LAN interface compared to a point to point interface. However, InterfaceLSPReceiveWindow is still very useful for the first LSPs sent and hence useful for the faster flooding in case of a single node failure which requires to flood a relatively small number of LSPs.

A compliant implementation may choose to not support this operation on a LAN interface.

5. Congestion control

Whereas flow control prevents the sender from overwhelming the receiver, congestion control prevents senders from overwhelming the network. For an IS-IS adjacency, the network between two IS-IS neighbors is relatively limited in scope and includes a single link which is typically over-sized compared to the capability of the IS-IS speakers. It also includes components inside both routers such as a fabric switch, line cards CPU and forwarding plane buffers which may experience congestion. This document proposes one optional congestion control algorithm but implementations may choose a different one or none.

The congestion control algorithm defined in this document is largely inspired by the TCP congestion control algorithm RFC 5681 [RFC5681]. A congestion control algorithm is comprised of three elements: a slow start phase, a congestion avoidance phase, and a transition between the two.

The proposed algorithm uses a variable Congestion window 'cwin'. It plays the same role as Receive Window described before. The main difference is that CWin is dynamically changed according to the feedback obtained by the PSNPs.

5.1. Slow start

The goal of the slow start phase is to grow fast and try to estimate the effective link capacity.

The algorithm is circuit scoped. At the beginning of the slow start, the sender starts with:

- o a congestion window (cwin) set to one. `cwin := 1;`
- o a number of acked LSPs. `acked_lsps := 0;`
- o a max seen bandwidth. `max_bw := 0;`

o a current rtt estimate. `cur_rtt := NA;`

Upon LSP sending, a sender records for said LSP the current time in `time_sent` and `acked_lsps` in `acked_lsps_sent`. This data is tied to each LSP.

Upon PSNP reception, a sender does the following:

```

cwin := min(cwin + nb_of_lsp_entries, rwin)
acked_lsps += nb_of_lsp_entries
max_diff := 0
max_bw := 0
for every LSP entry:
    time_to_ack := time_now - time_sent
    nb_acked := acked_lsps - acked_lsps_sent
    bw_est := nb_acked/time_to_ack
    max_bw := max(max_bw, bw_est)
    max_diff := max(max_diff, time_to_ack)

if cur_rtt == NA then cur_rtt = max_diff
else cur_rtt := 7/8 * cur_rtt + 1/8 * max_diff

```

Figure 1

Starting with the first PSNP, `max_bw` is checked every `cur_rtt`. Once it has stalled for 3 consecutive times, the congestion control algorithm transitions from slow start to congestion avoidance. There is bandwidth stalling when the bandwidth has not increased by at least 25% compared the last RTT. Note that this is similar to Google's BBR ([I-D.cardwell-iccr-g-bbr-congestion-control]) slow start phase.

5.2. Congestion avoidance

The goal of the congestion avoidance phase is to try to stay close to the effective capacity of the link. For this, the algorithm estimates the maximum time taken by the receiver to acknowledge a LSP. If an LSP arrives slower than this delay, congestion is inferred and `cwin` is decreased.

Upon PSNP reception, a sender does the following:

```
cwin = min(cwin + N/congestion window, rwin)
rtt_est := 0
for every LSP entry:
    time_to_ack = time_now - time_sent
    rtt_est = max(rtt_est, time_to_ack)

if rtt_var == NA then rtt_var = rtt_est / 2
else rtt_var = 3/4 * rtt_var + 1/4 * abs(cur_rtt - rtt_est)

cur_rtt = 7/8 * cur_rtt + 1/8 * rtt_est
```

Figure 2

Every LSP is checked to be acked within $\text{cur_rtt} + \text{rtt_var}$. If an LSP arrives late, cwin is divided by two. This behaviour is similar to TCP retransmission timer defined in RFC 6298 [RFC6298]

Note: there is no need for a timer per LSP. A timer per RTT is enough. During an RTT, sent LSPs are recorded in a list `list_1`. Once the RTT is over, `list_1` is kept and another list `list_2` is used to store the next LSPs. LSPs are removed from the lists when acked. At the end of the second RTT, every LSP in `list_1` should have been acked, so `list_1` is checked to be empty. `list_1` can then be reused for the next RTT.

If there is no transmitted LSP for a fixed period of time, e.g. 2 seconds, the sender switches back to the slow start phase.

5.3. Remarks

This algorithm's performance is dependent on the LPP value. Indeed, the smaller LPP is, the more information is available for the congestion control algorithm to perform well. However, it also increases the resources spent on sending PSNPs, so a tradeoff must be made. This document recommends to use an LPP of 15 or less.

Note that this congestion control algorithm benefits from the extensions proposed in this document. The advertisement of a receive window from the receiver (Section 4) avoids the use of an arbitrary maximum value by the sender. The faster acknowledgment of LSP (Section 4.2) allows for a faster control loop and hence a faster increase of the congestion window in the absence of congestion.

6. Interaction with other LSP rate limiting mechanisms

[ISO10589] describes a mechanism that limits the rate at which LSPs from the same source system are sent out on interfaces. (See the description of the parameter

minimumBroadcastLSPTranLSPTransmissionInterval in section 7.3.15.6 of [ISO10589]). In practice, however, router vendors have implemented mechanisms that limit the rate of LSPs sent on a given interface. This is often configurable on a per-interface basis using 'lsp-interval' or 'lsp-pacing-interval' CLI configuration). The mechanism described in the current document extends the practice of limiting the rate of LSPs sent on a given interface, by using parameters advertised by the LSP receiver. When the mechanism described in the current document is used, the mechanism described in section 7.3.15.6 of [ISO10589] is not used.

7. Determining values to be advertised in the Flooding Parameters TLV

The values that a receiving IS advertises do not need to be close to perfection. It is OK to be too low and hence not to use the full bandwidth or CPU resources. It is OK to be too high during some situation and hence have the receiver drop some LSPs as the IS-IS protocol has mechanisms to recover. What is not OK is to flood multiple order of magnitudes slower than both nodes can achieve, or to consistently overload the receiver.

The values may not need to be dynamic as a form of dynamic is provided by the dynamic acknowledgment of LSPs in SNP messages. Acknowledgments provides a feedback loop on how fast/slower the LSPs are processed by the receiver. They also signal that the LSPs have been processed by the receiver hence removed from receive window, explicitly signaling to the sender that more LSPs may be sent. By advertising relatively static parameters, we expect to produce overall flooding behavior similar to what might be achieved by manually configuring per-interface LSP rate limiting on all interfaces in the network. The advertised values may be based, for example, on an off line tests of the overall LSP processing speed for a particular set of hardware and the number of interfaces configured for IS-IS. With such a formula, the values advertised in the Flooding Parameters TLV would only change when additional IS-IS interfaces are configured.

The values may be updated dynamically, to reflect the relative change of load of the receiver, by improving the values when the receiver load is getting lower and degrading the values when the receiver load is getting higher. For example, if LSPs are regularly dropped, or the queue regularly comes close to being filled, then values may be too high. On the other hand, if the queue is barely used (by IS-IS), then values may be too low.

The values may also be absolute value reflecting relevant (averaged) hardware resources that are been monitored, typically the amount of buffer space used by incoming LSPs. In this case, care must be taken

when choosing the parameters influencing the values, in order to avoid undesirable or instable feedback loops. It would be undesirable to use a formula that depends, for example, on an active measurement of the instantaneous CPU load to modify the values advertised in the Flooding Parameters TLV. This could introduce feedback into the IGP flooding process that could produce unexpected behavior.

8. Operation considerations

As discussed in Section 4.3 , the solution is more effective on point to point adjacencies. Hence a broadcast interface (e.g. Ethernet) only shared by two IS-IS neighbors should be configured as point to point in order to have a more effective flooding.

9. IANA Considerations

IANA is requested to allocate one TLV from the IS-IS TLV codepoint registry.

Type	Description	IIH	LSP	SNP	Purge
----	-----	---	---	---	---
TBD1	Flooding Parameters TLV	y	n	n	n

Figure 3

This document creates the following sub-TLV Registry:

Name: Sub-TLVs for TLV TBD1 (Flooding Parameters TLV).

Registration Procedure: Expert Review [RFC8126] .

Type	Description
0	Reserved
1	InterfaceLSPReceiveWindow
2	minimumInterfaceLSPTransmissionInterval
3-255	Unassigned

Table 1: Initial allocations

10. Security Considerations

Any new security issues raised by the procedures in this document depend upon the ability of an attacker to inject a false but

apparently valid SNP or IIH, the ease/difficulty of which has not been altered.

As with others TLV advertisements, the use of a cryptographic authentication as defined in [RFC5304] or [RFC5310] allows the authentication of the peer and the integrity of the message. As this document defines a TLV for SNP or IIH message, the relevant cryptographic authentication is for SNP and IIH message.

In the absence of cryptographic authentication, as IS-IS does not run over IP but directly over the link layer, it's considered difficult to inject false SNP/IIH without having access to the link layer.

If a false SNP/IIH is sent with a Flooding Parameters TLV set to conservative values, the attacker can reduce the flooding speed between the two adjacent neighbors which can result in LSDB inconsistencies and transient forwarding loops. However, it is not significantly different than filtering or altering LSPDUs which would also be possible with access to the link layer. In addition, if the downstream flooding neighbor has multiple IGP neighbors, which is typically the case for reliability or topological reasons, it would receive LSPs at a regular speed from its other neighbors and hence would maintain LSDB consistency.

If a false SNP/IIH is sent with a Flooding Parameters TLV set to aggressive values, the attacker can increase the flooding speed which can either overload a node or more likely generate loss of LSPs. However, it is not significantly different than sending many LSPs which would also be possible with access to the link layer, even with cryptographic authentication enabled. In addition, IS-IS has procedures to detect the loss of LSPs and recover.

This TLV advertisement is not flooded across the network but only sent between adjacent IS-IS neighbors. This would limit the consequences in case of forged messages, and also limits the dissemination of such information.

11. Acknowledgments

The authors would like to thank Henk Smit, Sarah Chen, Xuesong Geng, Pierre Francois and Hannes Gredler for their reviews, comments and suggestions.

The authors would like to thank David Jacquet, Sarah Chen, and Qiangzhou Gao for the tests performed on commercial implementations and their identification of some limiting factors.

12. References

12.1. Normative References

- [ISO10589]
International Organization for Standardization,
"Intermediate system to Intermediate system intra-domain
routing information exchange protocol for use in
conjunction with the protocol for providing the
connectionless-mode Network Service (ISO 8473)", ISO/
IEC 10589:2002, Second Edition, Nov 2002.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119,
DOI 10.17487/RFC2119, March 1997,
<<https://www.rfc-editor.org/info/rfc2119>>.
- [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>>.
- [RFC6298] Paxson, V., Allman, M., Chu, J., and M. Sargent,
"Computing TCP's Retransmission Timer", RFC 6298,
DOI 10.17487/RFC6298, June 2011,
<<https://www.rfc-editor.org/info/rfc6298>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for
Writing an IANA Considerations Section in RFCs", BCP 26,
RFC 8126, DOI 10.17487/RFC8126, June 2017,
<<https://www.rfc-editor.org/info/rfc8126>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,
May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

12.2. Informative References

- [I-D.cardwell-iccr-g-bbr-congestion-control]
Cardwell, N., Cheng, Y., Yeganeh, S. H., and V. Jacobson,
"BBR Congestion Control", draft-cardwell-iccr-g-bbr-
congestion-control-00 (work in progress), July 2017.

[I-D.ietf-lsr-dynamic-flooding]

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

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <<https://www.rfc-editor.org/info/rfc793>>.

[RFC5681] Allman, M., Paxson, V., and E. Blanton, "TCP Congestion Control", RFC 5681, DOI 10.17487/RFC5681, September 2009, <<https://www.rfc-editor.org/info/rfc5681>>.

Appendix A. Changes / Author Notes

[RFC Editor: Please remove this section before publication]

00: Initial version.

01: Two notes added in section 3 "Operation".

02: Refresh, no technical change.

03:

- o Flooding Parameters TLV: name changed, advertised in both Hello and SNP rather than just Hello, contains sub-TLVs, parameters encoded in 4 octets.
- o Terminology: upstream/downstream terms removed, in favor of terms from ISO specification (transmitter, receiver); burst-size rename to receive-window.
- o Significant editorials changes.
- o New section on the faster acknowledgment of LSPs.
- o New section on the faster retransmission of lost LSPs.

04:

- o Adding general introduction on flow control, congestion control, loss detection and recovery.
- o Reorganizing sections as per the high level functions: flow control, congestion control, loss detection and recovery.

- o Adding a section on congestion control.

05:

- o Some editorials changes.
- o Updating section "Faster acknowledgments of LSPs" following the IS-IS flooding performance tests presented during IETF 108.
- o Updated IANA section (new registry).

06: Refresh, no technical change.

07:

- o Precision that if a LSP is lost and resent, it does not count twice in the InterfaceLSPReceiveWindow.
- o Title changed.
- o Removed fast retransmissions of LSPs.
- o Changed congestion control algorithm.
- o Removed support of TLV in SNP.

Authors' Addresses

Bruno Decraene
Orange

Email: bruno.decraene@orange.com

Chris Bowers
Juniper Networks, Inc.
1194 N. Mathilda Avenue
Sunnyvale, CA 94089
USA

Email: cbowers@juniper.net

Jayesh J
Juniper Networks, Inc.
1194 N. Mathilda Avenue
Sunnyvale, CA 94089
USA

Email: jayeshj@juniper.net

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

Email: tony.li@tony.li

Gunter Van de Velde
Nokia
Copernicuslaan 50
Antwerp 2018
Belgium

Email: gunter.van_de_velde@nokia.com

Guillaume Solignac
Orange

Email: guillaume.solignac@orange.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

Paul Wells
Cisco Systems

Email: pauwells@cisco.com

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

Email: tony.li@tony.li

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

Email: prz@juniper.net

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

Email: shraddha@juniper.net

LSR Working Group
Internet-Draft
Intended status: Standards Track
Expires: November 29, 2020

X. Xu
Alibaba Inc
S. Kini

P. Psena
C. Filsfils
S. Litkowski
Cisco Systems, Inc.
M. Bocci
Nokia
May 28, 2020

Signaling Entropy Label Capability and Entropy Readable Label Depth
Using IS-IS
draft-ietf-isis-mpls-elc-13

Abstract

Multiprotocol Label Switching (MPLS) has defined a mechanism to load-balance traffic flows using Entropy Labels (EL). An ingress Label Switching Router (LSR) cannot insert ELs for packets going into a given Label Switched Path (LSP) unless an egress LSR has indicated via signaling that it has the capability to process ELs, referred to as the Entropy Label Capability (ELC), on that LSP. In addition, it would be useful for ingress LSRs to know each LSR's capability for reading the maximum label stack depth and performing EL-based load-balancing, referred to as Entropy Readable Label Depth (ERLD). This document defines a mechanism to signal these two capabilities using IS-IS and BGP-LS.

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 November 29, 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. Advertising ELC Using IS-IS	3
4. Advertising ERLD Using IS-IS	4
5. Signaling ELC and ERLD in BGP-LS	4
6. IANA Considerations	4
7. Security Considerations	5
8. Contributors	5
9. Acknowledgements	6
10. References	6
10.1. Normative References	6
10.2. Informative References	7
Authors' Addresses	7

1. Introduction

[RFC6790] describes a method to load-balance Multiprotocol Label Switching (MPLS) traffic flows using Entropy Labels (EL). It also introduces the concept of Entropy Label Capability (ELC) and defines the signaling of this capability via MPLS signaling protocols. Recently, mechanisms have been defined to signal labels via link-state Interior Gateway Protocols (IGP) such as IS-IS [RFC8667]. This draft defines a mechanism to signal the ELC using IS-IS.

In cases where Segment Routing (SR) is used with the MPLS Data Plane (e.g., SR-MPLS [RFC8660]), it would be useful for ingress LSRs to know each intermediate LSR's capability of reading the maximum label stack depth and performing EL-based load-balancing. This capability, referred to as Entropy Readable Label Depth (ERLD) as defined in [RFC8662], may be used by ingress LSRs to determine the position of the EL label in the stack, and whether it's necessary to insert

multiple ELs at different positions in the label stack. This document defines a mechanism to signal the ERLD using IS-IS.

2. Terminology

This memo makes use of the terms defined in [RFC6790], and [RFC8662].

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

3. Advertising ELC Using IS-IS

Even though ELC is a property of the node, in some cases it is advantageous to associate and advertise the ELC with a prefix. In a multi-area network, routers may not know the identity of the prefix originator in a remote area, or may not know the capabilities of such originator. Similarly, in a multi-domain network, the identity of the prefix originator and its capabilities may not be known to the ingress LSR.

Bit 3 in the Prefix Attribute Flags [RFC7794] is used as the ELC Flag (E-flag), as shown in Figure 1. If a router has multiple interfaces, the router MUST NOT announce the ELC for any local host prefixes unless all of its interfaces are capable of processing ELs. If a router supports ELs on all of its interfaces, it SHOULD set the ELC for every local host prefix it advertises in IS-IS.

```

  0 1 2 3 4 5 6 7...
+--+--+--+--+--+--+...
|X|R|N|E|      ...
+--+--+--+--+--+--+...

```

Figure 1: Prefix Attribute Flags

E-flag: ELC Flag (Bit 3) - Set for local host prefix of the originating node if it supports ELC on all interfaces.

The ELC signaling MUST be preserved when a router propagates a prefix between ISIS levels [RFC5302].

When redistributing a prefix between two IS-IS protocol instances or redistributing from another protocol to an IS-IS protocol instance, a router SHOULD preserve the ELC signaling for that prefix if it exists. The exact mechanism used to exchange ELC between protocol instances running on an Autonomous System Boundary Router is outside of the scope of this document.

4. Advertising ERLD Using IS-IS

A new MSD-Type [RFC8491], called ERLD-MSD, is defined to advertise the ERLD [RFC8662] of a given router. A MSD-Type code 2 has been assigned by IANA for ERLD-MSD. The MSD-Value field is set to the ERLD in the range between 0 to 255. The scope of the advertisement depends on the application. If a router has multiple interfaces with different capabilities of reading the maximum label stack depth, the router MUST advertise the smallest value found across all its interfaces.

The absence of ERLD-MSD advertisements indicates only that the advertising node does not support advertisement of this capability.

The considerations for advertising the ERLD are specified in [RFC8662].

If the ERLD-MSD Type is received in the Link MSD Sub-TLV, it MUST be ignored.

5. Signaling ELC and ERLD in BGP-LS

The IS-IS extensions defined in this document can be advertised via BGP-LS (Distribution of Link-State and TE Information Using BGP) [RFC7752] using existing BGP-LS TLVs.

The ELC is advertised using the Prefix Attribute Flags TLV as defined in [I-D.ietf-idr-bgp-ls-segment-routing-ext].

The ERLD-MSD is advertised using the Node MSD TLV as defined in [I-D.ietf-idr-bgp-ls-segment-routing-msd].

6. IANA Considerations

Early allocation has been done by IANA for this document as follows:

- Bit 3 in the Bit Values for Prefix Attribute Flags Sub-TLV registry has been assigned to the ELC Flag. IANA is asked to update the registry to reflect the name used in this document: ELC Flag (E-flag).
- Type 2 in the IGP MSD-Types registry has been assigned for the ERLD-MSD. IANA is asked to update the registry to reflect the name used in this document: ERLD-MSD.

7. Security Considerations

This document specifies the ability to advertise additional node capabilities using IS-IS and BGP-LS. As such, the security considerations as described in [RFC7981], [RFC7752], [RFC7794], [RFC8491], [RFC8662], [I-D.ietf-idr-bgp-ls-segment-routing-ext] and [I-D.ietf-idr-bgp-ls-segment-routing-msd] are applicable to this document.

Incorrectly setting the E flag during origination, propagation or redistribution may lead to poor or no load-balancing of the MPLS traffic or black-holing of the MPLS traffic on the egress node.

Incorrectly setting of the ERLD value may lead to poor or no load-balancing of the MPLS traffic.

8. Contributors

The following people contributed to the content of this document and should be considered as co-authors:

Gunter Van de Velde (editor)
Nokia
Antwerp
BE

Email: gunter.van_de_velde@nokia.com

Wim Henderickx
Nokia
Belgium

Email: wim.henderickx@nokia.com

Keyur Patel
Arrcus
USA

Email: keyur@arrcus.com

9. Acknowledgements

The authors would like to thank Yimin Shen, George Swallow, Acee Lindem, Les Ginsberg, Ketan Talaulikar, Jeff Tantsura, Bruno Decraene Carlos Pignataro, Wim Hendrickx, and Gunter Van De Velde for their valuable comments.

10. References

10.1. Normative References

- [I-D.ietf-idr-bgp-ls-segment-routing-ext]
Previdi, S., Talaulikar, K., Filsfils, C., Gredler, H.,
and M. Chen, "BGP Link-State extensions for Segment
Routing", draft-ietf-idr-bgp-ls-segment-routing-ext-16
(work in progress), June 2019.
- [I-D.ietf-idr-bgp-ls-segment-routing-msd]
Tantsura, J., Chunduri, U., Talaulikar, K., Mirsky, G.,
and N. Triantafyllis, "Signaling MSD (Maximum SID Depth)
using Border Gateway Protocol - Link State", draft-ietf-
idr-bgp-ls-segment-routing-msd-18 (work in progress), May
2020.
- [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>>.
- [RFC5302] Li, T., Smit, H., and T. Przygienda, "Domain-Wide Prefix
Distribution with Two-Level IS-IS", RFC 5302,
DOI 10.17487/RFC5302, October 2008,
<<https://www.rfc-editor.org/info/rfc5302>>.
- [RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and
L. Yong, "The Use of Entropy Labels in MPLS Forwarding",
RFC 6790, DOI 10.17487/RFC6790, November 2012,
<<https://www.rfc-editor.org/info/rfc6790>>.
- [RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and
S. Ray, "North-Bound Distribution of Link-State and
Traffic Engineering (TE) Information Using BGP", RFC 7752,
DOI 10.17487/RFC7752, March 2016,
<<https://www.rfc-editor.org/info/rfc7752>>.

- [RFC7794] Ginsberg, L., Ed., Decraene, B., Previdi, S., Xu, X., and U. Chunduri, "IS-IS Prefix Attributes for Extended IPv4 and IPv6 Reachability", RFC 7794, DOI 10.17487/RFC7794, March 2016, <<https://www.rfc-editor.org/info/rfc7794>>.
- [RFC7981] Ginsberg, L., Previdi, S., and M. Chen, "IS-IS Extensions for Advertising Router Information", RFC 7981, DOI 10.17487/RFC7981, October 2016, <<https://www.rfc-editor.org/info/rfc7981>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8491] Tantsura, J., Chunduri, U., Aldrin, S., and L. Ginsberg, "Signaling Maximum SID Depth (MSD) Using IS-IS", RFC 8491, DOI 10.17487/RFC8491, November 2018, <<https://www.rfc-editor.org/info/rfc8491>>.
- [RFC8662] Kini, S., Kompella, K., Sivabalan, S., Litkowski, S., Shakir, R., and J. Tantsura, "Entropy Label for Source Packet Routing in Networking (SPRING) Tunnels", RFC 8662, DOI 10.17487/RFC8662, December 2019, <<https://www.rfc-editor.org/info/rfc8662>>.

10.2. Informative References

- [RFC8660] Bashandy, A., Ed., Filsfils, C., Ed., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with the MPLS Data Plane", RFC 8660, DOI 10.17487/RFC8660, December 2019, <<https://www.rfc-editor.org/info/rfc8660>>.
- [RFC8667] Previdi, S., Ed., Ginsberg, L., Ed., Filsfils, C., Bashandy, A., Gredler, H., and B. Decraene, "IS-IS Extensions for Segment Routing", RFC 8667, DOI 10.17487/RFC8667, December 2019, <<https://www.rfc-editor.org/info/rfc8667>>.

Authors' Addresses

Xiaohu Xu
Alibaba Inc

Email: xiaohu.xxh@alibaba-inc.com

Sriganesh Kini

Email: sriganeshkini@gmail.com

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

Email: ppsenak@cisco.com

Clarence Filsfils
Cisco Systems, Inc.
Brussels
Belgium

Email: cfilsfil@cisco.com

Stephane Litkowski
Cisco Systems, Inc.
La Rigourdiere
Cesson Sevigne
France

Email: slitkows@cisco.com

Matthew Bocci
Nokia
Shoppenhangers Road
Maidenhead, Berks
UK

Email: matthew.bocci@nokia.com

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

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

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

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on 13 August 2022.

Copyright Notice

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

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

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

Table of Contents

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

1. Overview

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

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

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

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

1.1. Requirements Language

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

1.2. Tree Diagrams

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

2. IS-IS MSD

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

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

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

2.1. IS-IS MSD YANG Module

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

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

  import ietf-isis {
    prefix isis;
  }

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

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

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

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

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

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

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


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

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

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

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

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

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

```
/* Node MSD TLV */
augment "/rt:routing/" +
    "rt:control-plane-protocols/rt:control-plane-protocol"+
    "/isis:isis/isis:database/isis:levels/isis:lsp"+
    "/isis:router-capabilities" {
    when "/rt:routing/rt:control-plane-protocols/"+
        "rt:control-plane-protocol/rt:type = 'isis:isis'" {
        description
            "This augment ISIS routing protocol when used";
    }
    description
        "This augments ISIS protocol LSDB router capability.";
    container node-msd-tlv {
        list node-msds {
            key "msd-type";
            leaf msd-type {
                type identityref {
                    base mpls-msd:msd-base-type;
                }
                description
                    "MSD-Types";
            }
            leaf msd-value {
                type uint8;
                description
                    "MSD value, in the range of 0-255.";
            }
            description
                "Node MSD is the smallest link MSD supported by
                the node.";
        }
        description
            "Node MSD is the number of SIDs supported by a node.";
        reference
            "RFC 8476: Signaling Maximum SID Depth (MSD) Using OSPF";
    }
}
```

```
/* link MSD sub-tlv */
augment "/rt:routing/" +
    "rt:control-plane-protocols/rt:control-plane-protocol"+
    "/isis:isis/isis:database/isis:levels/isis:lsp"+
    "/isis:extended-is-neighbor/isis:neighbor" {
    when "/rt:routing/rt:control-plane-protocols/"+
        "rt:control-plane-protocol/rt:type = 'isis:isis'" {
        description
            "This augment ISIS routing protocol when used";
    }
}
```

```

    }
    description
      "This augments ISIS protocol LSDB neighbor with
      Link MSD sub-TLV.";

    uses link-msd-sub-tlv;
  }

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

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

```

3. IS-IS Segment Routing

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

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

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

```

module: ietf-isis-sr
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/isis:isis:
    +--rw segment-routing
    |   +--rw enabled?      boolean
    |   +--rw bindings
    |       +--rw advertise
    |       |   +--rw policies*  string

```

```

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

```

```

    | +---ro local-block* []
    |   +---ro range-size?   uint32
    |   +---ro sid-sub-tlv
    |       +---ro sid?   uint32
+---ro srms-preference
    +---ro preference?   uint8
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/isis:isis/isis:database/isis:levels
    /isis:lsp/isis:extended-is-neighbor/isis:neighbor:
+---ro sid-list* [value]
    +---ro adj-sid-flags
    | +---ro bits*   identityref
    +---ro weight?   uint8
    +---ro neighbor-id?   isis:system-id
    +---ro value      uint32
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/isis:isis/isis:database
    /isis:levels/isis:lsp/isis:mt-is-neighbor/isis:neighbor:
+---ro sid-list* [value]
    +---ro adj-sid-flags
    | +---ro bits*   identityref
    +---ro weight?   uint8
    +---ro neighbor-id?   isis:system-id
    +---ro value      uint32
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/isis:isis/isis:database
    /isis:levels/isis:lsp/isis:extended-ipv4-reachability
    /isis:prefixes:
+---ro sid-list* [value]
    +---ro prefix-sid-flags
    | +---ro bits*   identityref
    +---ro algorithm?   uint8
    +---ro value      uint32
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/isis:isis/isis:database
    /isis:levels/isis:lsp/isis:mt-extended-ipv4-reachability
    /isis:prefixes:
+---ro sid-list* [value]
    +---ro prefix-sid-flags
    | +---ro bits*   identityref
    +---ro algorithm?   uint8
    +---ro value      uint32
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/isis:isis/isis:database
    /isis:levels/isis:lsp/isis:ipv6-reachability/isis:prefixes:
+---ro sid-list* [value]
    +---ro prefix-sid-flags
    | +---ro bits*   identityref

```

```

    +--ro algorithm?          uint8
    +--ro value                uint32
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/isis:isis/isis:database
    /isis:levels/isis:lsp/isis:mt-ipv6-reachability/isis:prefixes:
    +--ro sid-list* [value]
    +--ro prefix-sid-flags
    |   +--ro bits* identityref
    +--ro algorithm?          uint8
    +--ro value                uint32
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/isis:isis/isis:database
    /isis:levels/isis:lsp:
    +--ro segment-routing-bindings* [fec range]
    +--ro fec                  string
    +--ro range                uint16
    +--ro sid-binding-flags
    |   +--ro bits* identityref
    +--ro binding
    +--ro prefix-sid
    +--ro sid-list* [value]
    +--ro prefix-sid-flags
    |   +--ro bits* identityref
    +--ro algorithm?          uint8
    +--ro value                uint32

```

3.1. IS-IS Segment Routing configuration

3.1.1. Segment Routing activation

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

3.1.2. Advertising mapping server policy

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

3.1.3. IP Fast reroute

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

3.2. IS-IS Segment Routing YANG Module

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

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

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

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

  import ietf-isis {
    prefix "isis";
  }

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

  organization
    "IETF LSR - LSR Working Group";
```

contact

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

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

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

Author: Yingzhen Qu
<mailto:yingzhen.qu@futurewei.com>

Author: Pushpasis Sarkar
<mailto:pushpasis.ietf@gmail.com>

Author: Ing-Wher Chen
<mailto:ingwherchen@mitre.org>

Author: Jeff Tantsura
<mailto:jefftant.ietf@gmail.com>

";

description

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

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

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

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

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

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

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


```
reference "RFC XXXX";

revision 2022-02-09 {
  description
    "Initial revision.";
  reference "RFC XXXX";
}

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

identity mpls-ipv4 {
  base sr-capability;
  description
    "If set, then the router is capable of
    processing SR MPLS encapsulated IPv4 packets
    on all interfaces.";
}

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

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

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

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

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

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

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

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

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

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

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

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

identity lo-bit {
```

```
    base adj-sid-bit;
    description
        "Local flag.";
}

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

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

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

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

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

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

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

identity a-bit {
```

```
    base sid-binding-bit;
    description
        "Attached flag.";
}

/* Features */

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

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

/* Groupings */

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

grouping sr-capability {
    description
        "SR capability grouping.";
    container sr-capability {
        description
            "Segment Routing capability.";
        container sr-capability {
            leaf-list sr-capability-bits {
                type identityref {
                    base sr-capability;
                }
            }
            description "SR Capability sub-tlv flags list.";
        }
    }
}
```

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

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

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

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

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

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

  list sid-list {
    key value;

    container prefix-sid-flags {
      leaf-list bits {
        type identityref {
          base prefix-sid-bit;
        }
        description
          "Prefix SID Sub-TLV flag bits list.";
      }
      description
        "Describes flags associated with the
        segment ID.";
    }

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

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

  list sid-list {
    key value;

    container adj-sid-flags {
      leaf-list bits {
        type identityref {
          base adj-sid-bit;
        }
      }
    }
  }
}
```

```
    }
    description "Adj sid sub-tlv flags list.";
  }
  description "Adj-sid sub-tlv flags.";
}

leaf weight {
  type uint8;
  description
    "The value represents the weight of the Adj-SID
    for the purpose of load balancing.";
}
leaf neighbor-id {
  type isis:system-id;
  description
    "Describes the system ID of the neighbor
    associated with the SID value. This is only
    used on LAN adjacencies.";
}
leaf value {
  type uint32;
  description
    "Value of the Adj-SID.";
}
description
  "List of segments.";
}
}

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

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

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

    container sid-binding-flags {
      leaf-list bits {
```



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

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

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

/* Cfg */

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

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

```
    }
  }

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

    uses sr-mpls:igp-interface;
  }

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

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

  augment "/rt:routing/" +
    "rt:control-plane-protocols/rt:control-plane-protocol"+
    "/isis:isis/isis:interfaces/isis:interface"+
    "/isis:fast-reroute/isis:lfa/isis:remote-lfa" {
    when "/rt:routing/rt:control-plane-protocols/" +
      "rt:control-plane-protocol/rt:type = 'isis:isis'" {
```

```
        description
          "This augment ISIS routing protocol when used";
      }
      description
        "This augments ISIS remoteLFA config with
         use of segment-routing path.";

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

  /* Operational states */

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

    uses adjacency-state;
  }

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

    uses sr-capability;
    uses sr-algorithm;
  }
```

```
    uses srlb;
    uses srms-preference;
  }

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

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

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

  augment "/rt:routing/" +
    "rt:control-plane-protocols/rt:control-plane-protocol"+
```

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

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

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

augment "/rt:routing/" +
    "rt:control-plane-protocols/rt:control-plane-protocol"+
    "/isis:isis/isis:database/isis:levels/isis:lsp" {
when "/rt:routing/rt:control-plane-protocols/"+
    "rt:control-plane-protocol/rt:type = 'isis:isis'" {
    description
        "This augment ISIS routing protocol when used";
    }
}
```

```
    description
      "This augments ISIS protocol LSDB.";
      uses segment-routing-binding-tlv;
  }

  /* Notifications */
}
<CODE ENDS>
```

4. Security Considerations

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

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

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

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

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

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

Some of the readable data nodes in the modules may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes.

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

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

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

And the augmentations to the ISIS link state database.

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

5. Contributors

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

6. Acknowledgements

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

7. IANA Considerations

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

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

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

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

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

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

8. Normative References

- [I-D.ietf-isis-yang-isis-cfg]
Litkowski, S., Yeung, D., Lindem, A., Zhang, J., and L. Lhotka, "YANG Data Model for IS-IS Protocol", Work in Progress, Internet-Draft, draft-ietf-isis-yang-isis-cfg-42, 15 October 2019, <<https://www.ietf.org/archive/id/draft-ietf-isis-yang-isis-cfg-42.txt>>.
- [I-D.ietf-spring-sr-yang]
Litkowski, S., Qu, Y., Sarkar, P., and J. Tantsura, "YANG Data Model for Segment Routing", Work in Progress, Internet-Draft, draft-ietf-spring-sr-yang-15, 28 December 2017, <<http://www.ietf.org/internet-drafts/draft-ietf-spring-sr-yang-15.txt>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <<https://www.rfc-editor.org/info/rfc6536>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.

- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8491] Tantsura, J., Chunduri, U., Aldrin, S., and L. Ginsberg, "Signaling Maximum SID Depth (MSD) Using IS-IS", RFC 8491, DOI 10.17487/RFC8491, November 2018, <<https://www.rfc-editor.org/info/rfc8491>>.
- [RFC8667] Previdi, S., Ed., Ginsberg, L., Ed., Filsfils, C., Bashandy, A., Gredler, H., and B. Decraene, "IS-IS Extensions for Segment Routing", RFC 8667, DOI 10.17487/RFC8667, December 2019, <<https://www.rfc-editor.org/info/rfc8667>>.

Authors' Addresses

Stephane Litkowski
Cisco Systems

Email: slitkows.ietf@gmail.com

Yingzhen Qu
Futurewei

Email: yingzhen.qu@futurewei.com

Pushpasis Sarkar
Individual

Email: pushpasis.ietf@gmail.com

Ing-Wher Chen
The MITRE Corporation

Email: ingwherchen@mitre.org

Jeff Tantsura
Microsoft

Email: jefftant.ietf@gmail.com

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

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

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

Abstract

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

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

Status of This Memo

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

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

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

This Internet-Draft will expire on 10 June 2022.

Copyright Notice

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

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

Table of Contents

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

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

1. Introduction

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

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

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

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

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

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

1.1. Requirements Language

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

2. Problem Statement

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

3. Solution Requirements

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

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

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

4. Dynamic Flooding

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

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

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

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

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

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

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

4.1. Applicability

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

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

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

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

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

4.2. Leader election

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

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

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

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

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

4.3. Computing the Flooding Topology

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

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

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

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

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

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

4.4. Topologies on Complete Bipartite Graphs

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

4.4.1. A Minimal Flooding Topology

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

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

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

4.4.2. Xia Topologies

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

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

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

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

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

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

4.4.3. Optimization

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

4.5. Encoding the Flooding Topology

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

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

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

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

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

4.6. Advertising the Local Edges Enabled for Flooding

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

The following guidelines apply:

Advertisement of LEEFs is optional.

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

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

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

5. Protocol Elements

5.1. IS-IS TLVs

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

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

5.1.1. IS-IS Area Leader Sub-TLV

The Area Leader Sub-TLV allows a system to:

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

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

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

The Area Leader Sub-TLV is advertised as a Sub-TLV of the IS-IS Router Capability TLV-242 that is defined in [RFC7981] and has the following format:

0										1										2										3																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+

Type: TBD1

Length: 2

Priority: 0-255, unsigned integer

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

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

5.1.2. IS-IS Dynamic Flooding Sub-TLV

The Dynamic Flooding Sub-TLV allows a system to:

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

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

The Dynamic Flooding Sub-TLV is advertised as a Sub-TLV of the IS-IS Router Capability TLV (242) [RFC7981] and has the following format:


```

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

```

Type: TBD7

Length: 0-255; number of Algorithms

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

5.1.3. IS-IS Area Node IDs TLV

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

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

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

The format of the Area Node IDs TLV is:

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Type      |      Length      | Starting Index |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|L| Reserved    | Node IDs ...
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
Node IDs continued ....
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Type: TBD2

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

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

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

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

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

5.1.1.4. IS-IS Flooding Path TLV

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

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

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

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

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

The Flooding Path TLV has the format:

0									1									2									3								
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1				
Type									Length									Starting Index																	
Index 2									Additional indices ...																										

Type: TBD3

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

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

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

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

5.1.5. IS-IS Flooding Request TLV

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

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

The Flooding Request TLV has the format:

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

Type: TBD9

Length: 1 + number of advertised Flooding Scopes

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

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

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

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

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

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

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

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

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

5.1.6. IS-IS LEEF Advertisement

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

The following bit value is defined by this document:

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

5.2. OSPF LSAs and TLVs

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

Following objects are added:

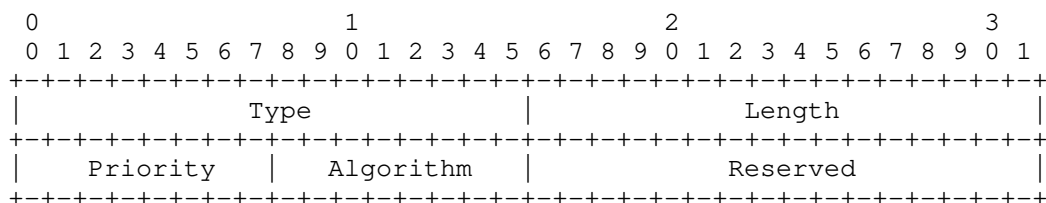
1. A TLV that is used to advertise the preference for becoming Area Leader.
2. A TLV that is used to indicate the support for Dynamic Flooding and the algorithms that the advertising node supports for distributed mode, if any.
3. OSPFv2 Opaque LSA and OSPFv3 LSA to advertise the flooding topology for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

5.2.1. OSPF Area Leader Sub-TLV

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

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

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



Type: TBD4

Length: 4 octets

Priority: 0-255, unsigned integer

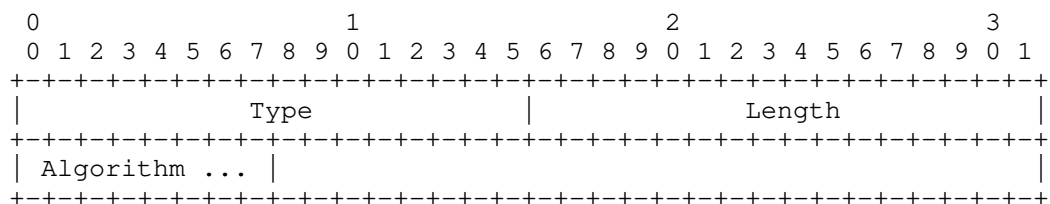
Algorithm: as defined in Section 5.1.1.

5.2.2. OSPF Dynamic Flooding Sub-TLV

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

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

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



Type: TBD8

Length: number of Algorithms

Algorithm: as defined in Section 5.1.1.

5.2.3. OSPFv2 Dynamic Flooding Opaque LSA

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

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

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

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

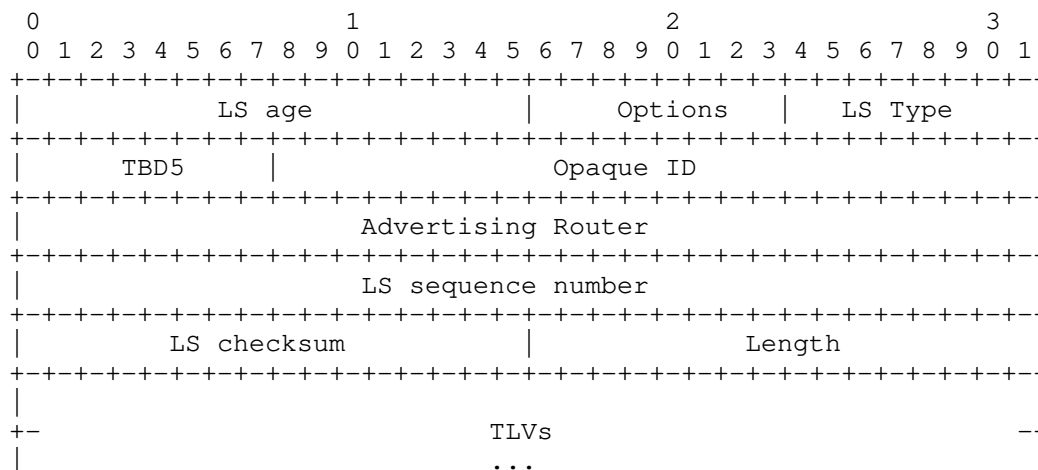


Figure 1: OSPFv2 Dynamic Flooding Opaque LSA

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

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

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

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

5.2.4. OSPFv3 Dynamic Flooding LSA

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

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

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

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

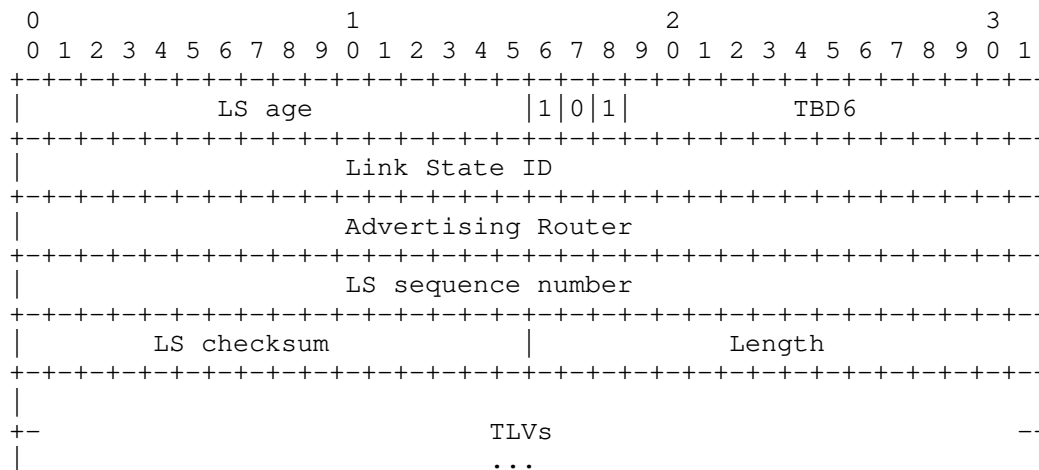


Figure 2: OSPFv3 Dynamic Flooding LSA

5.2.5. OSPF Area Router ID TLVs

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

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

5.2.5.1. OSPFv2 Area Router ID TLV

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

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

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

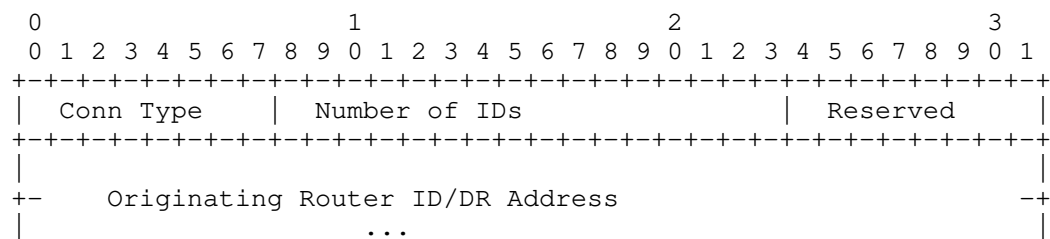


Figure 3: OSPFv2 Router IDs TLV Entry

Conn Type: 1 byte

- The following values are defined:

- 1 - Router
- 2 - Designated Router

Number of IDs: 2 bytes

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

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

The format of the Area Router IDs TLV is:

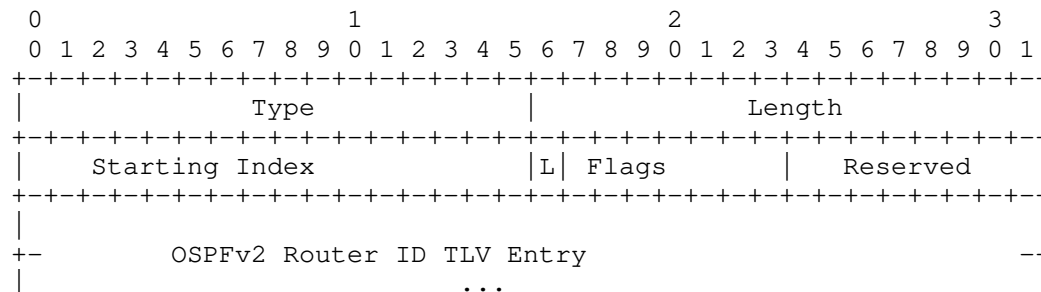


Figure 4: OSPFv2 Area Router IDs TLV

TLV Type: 1

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

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

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

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

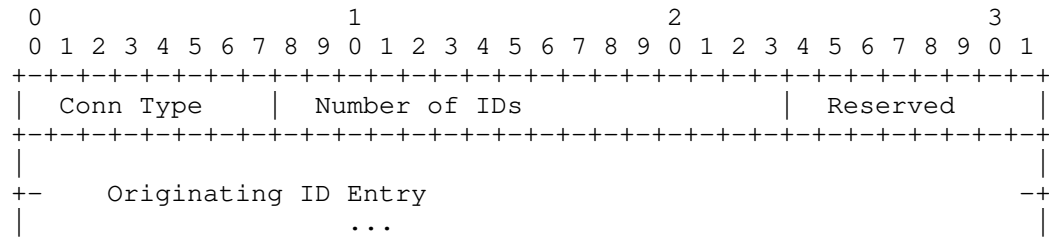
If there are multiple OSPFv2 Area Router ID TLVs with the L bit set advertised by the same router, the TLV which specifies the smaller maximum index is used and the other TLV(s) with L bit set are ignored. TLVs which specify Router IDs with indices greater than that specified by the TLV with the L bit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

5.2.5.2. OSPFv3 Area Router ID TLV

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

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

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



where

Conn Type - 1 byte

The following values are defined:

1 - Router

2 - Designated Router

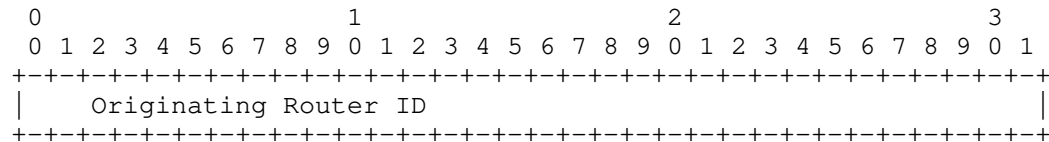
Number of IDs - 2 bytes

Reserved - 1 byte

MUST be transmitted as 0 and MUST be ignored on receipt

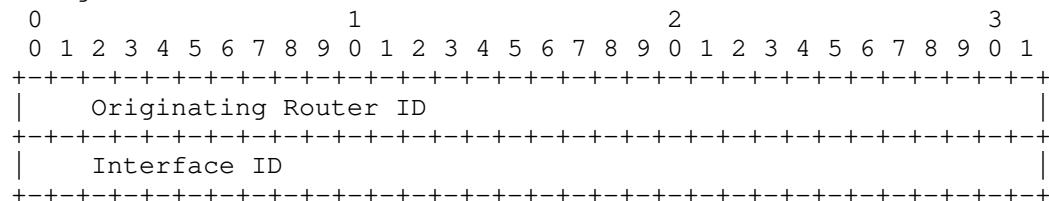
Originating ID Entry takes one of the following forms:

Router:



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

Designated Router:



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

Figure 5: OSPFv3 Router ID TLV Entry

The format of the OSPFv3Area Router IDs TLV is:

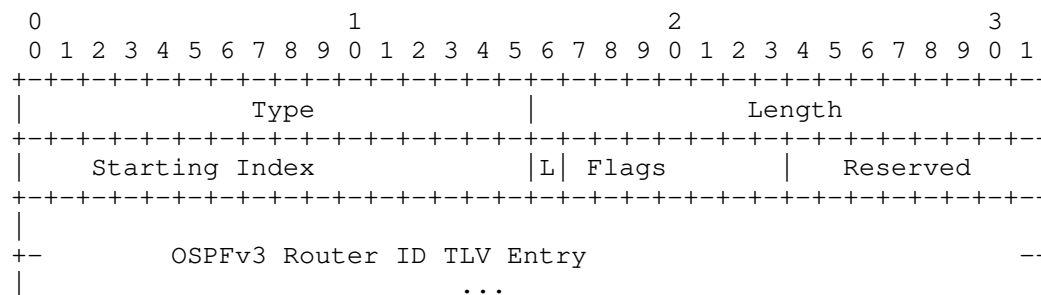


Figure 6: OSPFv3 Area Router IDs TLV

TLV Type: 1

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

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

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

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

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

5.2.6. OSPF Flooding Path TLV

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

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

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

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

The Flooding Path TLV has the format:

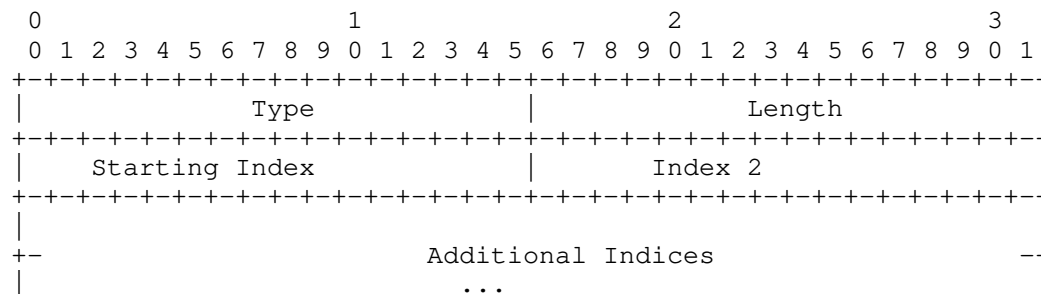


Figure 7: OSPF Flooding Path TLV

TLV Type: 2

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

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

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

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

5.2.7. OSPF Flooding Request Bit

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

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

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

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

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

5.2.8. OSPF LEEF Advertisement

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

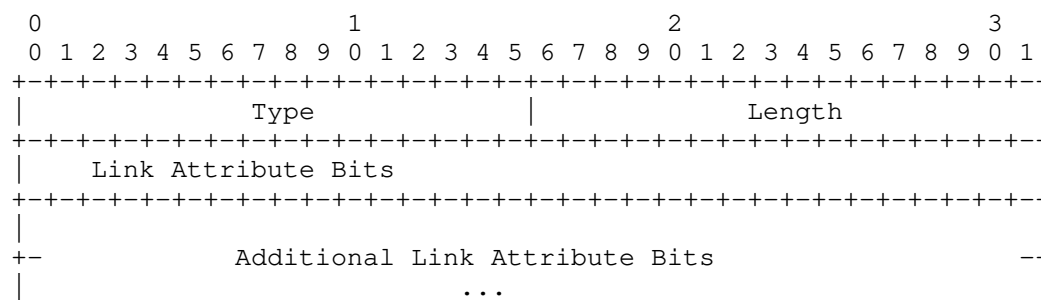


Figure 8: OSPF Link Attributes Bits TLV

Type: TBD and specific to OSPFv2 and OSPFv3

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

The following bits are defined:

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

OSPF Link-attribute Bits TLV appears as:

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

6. Behavioral Specification

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

6.1. Terminology

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

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

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

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

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

6.2. Flooding Topology

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

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

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

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

6.3. Leader Election

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

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

6.4. Area Leader Responsibilities

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

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

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

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

6.5. Distributed Flooding Topology Calculation

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

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

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

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

6.6. Use of LANs in the Flooding Topology

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

6.6.1. Use of LANs in Centralized mode

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

6.6.2. Use of LANs in Distributed Mode

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

6.6.2.1. Partial flooding on a LAN in IS-IS

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

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

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

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

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

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

6.6.2.2. Partial Flooding on a LAN in OSPF

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

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

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

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

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

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

6.7. Flooding Behavior

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

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

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

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

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

6.8. Treatment of Topology Events

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

6.8.1. Temporary Addition of Link to Flooding Topology

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

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

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

Flooding is enabled locally on the link.

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

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

- Node itself is connected to the current flooding topology.

- Adjacent node is connected to the current flooding topology.

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

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

6.8.2. Local Link Addition

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

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

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

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

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

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

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

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

6.8.3. Node Addition

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

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

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

6.8.4. Failures of Link Not on Flooding Topology

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

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

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

6.8.5. Failures of Link On the Flooding Topology

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

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

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

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

6.8.6. Node Deletion

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

6.8.7. Local Link Addition to the Flooding Topology

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

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

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

6.8.8. Local Link Deletion from the Flooding Topology

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

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

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

6.8.9. Treatment of Disconnected Adjacent Nodes

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

6.8.10. Failure of the Area Leader

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

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

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

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

6.8.11. Recovery from Multiple Failures

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

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

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

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

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

6.8.12. Rate Limiting Temporary Flooding

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

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

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

7. IANA Considerations

7.1. IS-IS

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

Type: TBD1

Description: IS-IS Area Leader Sub-TLV

Reference: This document (Section 5.1.1)

Type: TBD7

Description: IS-IS Dynamic Flooding Sub-TLV

Reference: This document (Section 5.1.2)

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

Type: TBD2

Description: IS-IS Area System IDs TLV

Reference: This document (Section 5.1.3)

Type: TBD3

Description: IS-IS Flooding Path TLV

Reference: This document (Section 5.1.4)

Type: TBD9

Description: IS-IS Flooding Request TLV

Reference: This document (Section 5.1.5)

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

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

7.2. OSPF

This document requests the following code points from the "OSPF Router Information (RI) TLVs" registry:

Type: TBD4

Description: OSPF Area Leader Sub-TLV

Reference: This document (Section 5.2.1)

Type: TBD8

Description: OSPF Dynamic Flooding Sub-TLV

Reference: This document (Section 5.2.2)

This document requests the following code point from the "Opaque Link-State Advertisements (LSA) Option Types" registry:

Type: TBD5

Description: OSPFv2 Dynamic Flooding Opaque LSA

Reference: This document (Section 5.2.3)

This document requests the following code point from the "OSPFv3 LSA Function Codes" registry:

Type: TBD6

Description: OSPFv3 Dynamic Flooding LSA

Reference: This document (Section 5.2.4)

This document requests a new bit in LLS Type 1 Extended Options and Flags registry:

Bit Position: TBD10

Description: Flooding Request bit

Reference: This document (Section 5.2.7)

This document requests the following code point from the "OSPFv2 Extended Link TLV Sub-TLVs" registry:

Type: TBD11

Description: OSPFv2 Link Attributes Bits Sub-TLV

Reference: This document (Section 5.2.8)

This document requests the following code point from the "OSPFv3 Extended LSA Sub-TLVs" registry:

Type: TBD12

Description: OSPFv3 Link Attributes Bits Sub-TLV

Reference: This document (Section 5.2.8)

7.2.1. OSPF Dynamic Flooding LSA TLVs Registry

This specification also requests a new registry - "OSPF Dynamic Flooding LSA TLVs". New values can be allocated via IETF Review or IESG Approval

The "OSPF Dynamic Flooding LSA TLVs" registry will define top-level TLVs for the OSPFv2 Dynamic Flooding Opaque LSA and OSPFv3 Dynamic Flooding LSAs. It should be added to the "Open Shortest Path First (OSPF) Parameters" registries group.

The following initial values are allocated:

Type: 0

Description: Reserved

Reference: This document

Type: 1

Description: OSPF Area Router IDs TLV

Reference: This document (Section 5.2.5)

Type: 2

Description: OSPF Flooding Path TLV

Reference: This document (Section 5.2.6)

Types in the range 32768-33023 are for experimental use; these will not be registered with IANA, and MUST NOT be mentioned by RFCs.

Types in the range 33024-65535 are not to be assigned at this time. Before any assignments can be made in the 33024-65535 range, there MUST be an IETF specification that specifies IANA Considerations that covers the range being assigned.

7.2.2. OSPF Link Attributes Sub-TLV Bit Values Registry

This specification also requests a new registry - "OSPF Link Attributes Sub-TLV Bit Values". New values can be allocated via IETF Review or IESG Approval

The "OSPF Link Attributes Sub-TLV Bit Values" registry defines Link Attribute bit values for the OSPFv2 Link Attributes Sub-TLV and OSPFv3 Link Attributes Sub-TLV. It should be added to the "Open Shortest Path First (OSPF) Parameters" registries group.

The following initial value is allocated:

Bit Number: 0

Description: Local Edge Enabled for Flooding(LEEF)

Reference: This document (Section 5.2.8)

7.3. IGP

IANA is requested to set up a registry called "IGP Algorithm Type For Computing Flooding Topology" under an existing "Interior Gateway Protocol (IGP) Parameters" IANA registries.

Values in this registry come from the range 0-255.

The initial values in the IGP Algorithm Type For Computing Flooding Topology registry are:

0: Reserved for centralized mode.

1-127: Available for standards action. Individual values are to be assigned according to the "Specification Required" policy defined in [RFC8126].

128-254: Reserved for private use.

255: Reserved.

8. Security Considerations

This document introduces no new security issues. Security of routing within a domain is already addressed as part of the routing protocols themselves. This document proposes no changes to those security architectures.

It is possible that an attacker could become Area Leader and introduce a flawed flooding algorithm into the network thus compromising the operation of the protocol. Authentication methods as describe in [RFC5304] and [RFC5310] for IS-IS, [RFC2328] and [RFC7474] for OSPFv2 and [RFC5340] and [RFC4552] for OSPFv3 SHOULD be used to prevent such attack.

9. Acknowledgements

The authors would like to thank Sarah Chen for her contribution to this work.

The authors would like to thank Zeqing (Fred) Xia, Naiming Shen, Adam Sweeney and Olufemi Komolafe for their helpful comments.

The authors would like to thank Tom Edsall for initially introducing them to the problem.

Advertising Local Edges Enabled for Flooding (LEEF) is based on an idea proposed in [I-D.cc-lsr-flooding-reduction]. We wish to thank the authors of that draft.

10. References

10.1. Normative References

- [ISO10589] International Organization for Standardization, "Intermediate System to Intermediate System Intra-Domain Routing Exchange Protocol for use in Conjunction with the Protocol for Providing the Connectionless-mode Network Service (ISO 8473)", ISO/IEC 10589:2002, October 2002.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

- [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <<https://www.rfc-editor.org/info/rfc2328>>.
- [RFC4552] Gupta, M. and N. Melam, "Authentication/Confidentiality for OSPFv3", RFC 4552, DOI 10.17487/RFC4552, June 2006, <<https://www.rfc-editor.org/info/rfc4552>>.
- [RFC5029] Vasseur, JP. and S. Previdi, "Definition of an IS-IS Link Attribute Sub-TLV", RFC 5029, DOI 10.17487/RFC5029, September 2007, <<https://www.rfc-editor.org/info/rfc5029>>.
- [RFC5250] Berger, L., Bryskin, I., Zinin, A., and R. Coltun, "The OSPF Opaque LSA Option", RFC 5250, DOI 10.17487/RFC5250, July 2008, <<https://www.rfc-editor.org/info/rfc5250>>.
- [RFC5304] Li, T. and R. Atkinson, "IS-IS Cryptographic Authentication", RFC 5304, DOI 10.17487/RFC5304, October 2008, <<https://www.rfc-editor.org/info/rfc5304>>.
- [RFC5310] Bhatia, M., Manral, V., Li, T., Atkinson, R., White, R., and M. Fanto, "IS-IS Generic Cryptographic Authentication", RFC 5310, DOI 10.17487/RFC5310, February 2009, <<https://www.rfc-editor.org/info/rfc5310>>.
- [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <<https://www.rfc-editor.org/info/rfc5340>>.
- [RFC7356] Ginsberg, L., Previdi, S., and Y. Yang, "IS-IS Flooding Scope Link State PDUs (LSPs)", RFC 7356, DOI 10.17487/RFC7356, September 2014, <<https://www.rfc-editor.org/info/rfc7356>>.
- [RFC7474] Bhatia, M., Hartman, S., Zhang, D., and A. Lindem, Ed., "Security Extension for OSPFv2 When Using Manual Key Management", RFC 7474, DOI 10.17487/RFC7474, April 2015, <<https://www.rfc-editor.org/info/rfc7474>>.
- [RFC7684] Psenak, P., Gredler, H., Shakir, R., Henderickx, W., Tantsura, J., and A. Lindem, "OSPFv2 Prefix/Link Attribute Advertisement", RFC 7684, DOI 10.17487/RFC7684, November 2015, <<https://www.rfc-editor.org/info/rfc7684>>.
- [RFC7770] Lindem, A., Ed., Shen, N., Vasseur, JP., Aggarwal, R., and S. Shaffer, "Extensions to OSPF for Advertising Optional Router Capabilities", RFC 7770, DOI 10.17487/RFC7770, February 2016, <<https://www.rfc-editor.org/info/rfc7770>>.

- [RFC7981] Ginsberg, L., Previdi, S., and M. Chen, "IS-IS Extensions for Advertising Router Information", RFC 7981, DOI 10.17487/RFC7981, October 2016, <<https://www.rfc-editor.org/info/rfc7981>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [RFC8362] Lindem, A., Roy, A., Goethals, D., Reddy Vallem, V., and F. Baker, "OSPFv3 Link State Advertisement (LSA) Extensibility", RFC 8362, DOI 10.17487/RFC8362, April 2018, <<https://www.rfc-editor.org/info/rfc8362>>.

10.2. Informative References

- [Clos] Clos, C., "A Study of Non-Blocking Switching Networks", The Bell System Technical Journal Vol. 32(2), DOI 10.1002/j.1538-7305.1953.tb01433.x, March 1953, <<http://dx.doi.org/10.1002/j.1538-7305.1953.tb01433.x>>.
- [I-D.cc-lsr-flooding-reduction]
Chen, H., Toy, M., Yang, Y., Wang, A., Liu, X., Fan, Y., and L. Liu, "Flooding Topology Computation Algorithm", Work in Progress, Internet-Draft, draft-cc-lsr-flooding-reduction-09, 5 June 2020, <<https://www.ietf.org/archive/id/draft-cc-lsr-flooding-reduction-09.txt>>.
- [Leiserson]
Leiserson, C. E., "Fat-Trees: Universal Networks for Hardware-Efficient Supercomputing", IEEE Transactions on Computers 34(10):892-901, 1985.
- [RFC2973] Balay, R., Katz, D., and J. Parker, "IS-IS Mesh Groups", RFC 2973, DOI 10.17487/RFC2973, October 2000, <<https://www.rfc-editor.org/info/rfc2973>>.
- [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, DOI 10.17487/RFC3630, September 2003, <<https://www.rfc-editor.org/info/rfc3630>>.
- [RFC4811] Nguyen, L., Roy, A., and A. Zinin, "OSPF Out-of-Band Link State Database (LSDB) Resynchronization", RFC 4811, DOI 10.17487/RFC4811, March 2007, <<https://www.rfc-editor.org/info/rfc4811>>.

[RFC7938] Lapukhov, P., Premji, A., and J. Mitchell, Ed., "Use of BGP for Routing in Large-Scale Data Centers", RFC 7938, DOI 10.17487/RFC7938, August 2016, <<https://www.rfc-editor.org/info/rfc7938>>.

Authors' Addresses

Tony Li (editor)
Juniper Networks
1133 Innovation Way
Sunnyvale, California 94089
United States of America

Email: tony.li@tony.li

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

Email: prz@juniper.net

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

Email: ppsenak@cisco.com

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

Email: ginsberg@cisco.com

Huaimo Chen
Futurewei
Boston, Ma,
United States of America

Email: hchen@futurewei.com

Dave Cooper
CenturyLink
1025 Eldorado Blvd
Broomfield, Colorado 80021
United States of America

Email: Dave.Cooper@centurylink.com

Luay Jalil
Verizon
Richardson, Texas 75081
United States of America

Email: luay.jalil@verizon.com

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

Email: sd947e@att.com

Gyan S. Mishra
Verizon Inc.
13101 Columbia Pike
Silver Spring, Maryland 20904
United States of America

Phone: 301 502-1347
Email: gyan.s.mishra@verizon.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 9, 2022

P. Psenak, Ed.
Cisco Systems
S. Hegde
Juniper Networks, Inc.
C. Filsfils
Cisco Systems, Inc.
K. Talaulikar
Arrcus, Inc
A. Gulko
Edward Jones
April 7, 2022

IGP Flexible Algorithm
draft-ietf-lsr-flex-algo-19

Abstract

IGP protocols traditionally compute best paths over the network based on the IGP metric assigned to the links. Many network deployments use RSVP-TE based or Segment Routing based Traffic Engineering to steer traffic over a path that is computed using different metrics or constraints than the shortest IGP path. This document proposes a solution that allows IGPs themselves to compute constraint-based paths over the network. This document also specifies a way of using Segment Routing (SR) Prefix-SIDs and SRv6 locators to steer packets along the constraint-based paths.

Status of This Memo

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

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

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

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

Copyright Notice

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

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

Table of Contents

1. Introduction	3
2. Requirements Language	4
3. Terminology	4
4. Flexible Algorithm	5
5. Flexible Algorithm Definition Advertisement	6
5.1. IS-IS Flexible Algorithm Definition Sub-TLV	6
5.2. OSPF Flexible Algorithm Definition TLV	8
5.3. Common Handling of Flexible Algorithm Definition TLV	9
6. Sub-TLVs of IS-IS FAD Sub-TLV	10
6.1. IS-IS Flexible Algorithm Exclude Admin Group Sub-TLV	11
6.2. IS-IS Flexible Algorithm Include-Any Admin Group Sub-TLV	12
6.3. IS-IS Flexible Algorithm Include-All Admin Group Sub-TLV	12
6.4. IS-IS Flexible Algorithm Definition Flags Sub-TLV	13
6.5. IS-IS Flexible Algorithm Exclude SRLG Sub-TLV	14
7. Sub-TLVs of OSPF FAD TLV	15
7.1. OSPF Flexible Algorithm Exclude Admin Group Sub-TLV	15
7.2. OSPF Flexible Algorithm Include-Any Admin Group Sub-TLV	16
7.3. OSPF Flexible Algorithm Include-All Admin Group Sub-TLV	16
7.4. OSPF Flexible Algorithm Definition Flags Sub-TLV	16
7.5. OSPF Flexible Algorithm Exclude SRLG Sub-TLV	17
8. IS-IS Flexible Algorithm Prefix Metric Sub-TLV	18
9. OSPF Flexible Algorithm Prefix Metric Sub-TLV	19
10. OSPF Flexible Algorithm ASBR Reachability Advertisement	21
10.1. OSPFv2 Extended Inter-Area ASBR LSA	21
10.1.1. OSPFv2 Extended Inter-Area ASBR TLV	23
10.2. OSPF Flexible Algorithm ASBR Metric Sub-TLV	23
11. Advertisement of Node Participation in a Flex-Algorithm	25
11.1. Advertisement of Node Participation for Segment Routing	26
11.2. Advertisement of Node Participation for Other Applications	26
12. Advertisement of Link Attributes for Flex-Algorithm	26

13. Calculation of Flexible Algorithm Paths	27
13.1. Multi-area and Multi-domain Considerations	29
14. Flex-Algorithm and Forwarding Plane	31
14.1. Segment Routing MPLS Forwarding for Flex-Algorithm	32
14.2. SRv6 Forwarding for Flex-Algorithm	32
14.3. Other Applications' Forwarding for Flex-Algorithm	33
15. Operational Considerations	33
15.1. Inter-area Considerations	33
15.2. Usage of SRLG Exclude Rule with Flex-Algorithm	34
15.3. Max-metric consideration	35
16. Backward Compatibility	35
17. Security Considerations	35
18. IANA Considerations	36
18.1. IGP IANA Considerations	36
18.1.1. IGP Algorithm Types Registry	36
18.1.2. IGP Metric-Type Registry	36
18.2. Flexible Algorithm Definition Flags Registry	37
18.3. IS-IS IANA Considerations	37
18.3.1. Sub TLVs for Type 242	37
18.3.2. Sub TLVs for for TLVs 135, 235, 236, and 237	37
18.3.3. Sub-Sub-TLVs for Flexible Algorithm Definition Sub-TLV	37
18.4. OSPF IANA Considerations	38
18.4.1. OSPF Router Information (RI) TLVs Registry	38
18.4.2. OSPFv2 Extended Prefix TLV Sub-TLVs	39
18.4.3. OSPFv3 Extended-LSA Sub-TLVs	39
18.4.4. OSPF Flex-Algorithm Prefix Metric Bits	39
18.4.5. OSPF Opaque LSA Option Types	39
18.4.6. OSPFv2 Extended Inter-Area ASBR TLVs	40
18.4.7. OSPFv2 Inter-Area ASBR Sub-TLVs	40
18.4.8. OSPF Flexible Algorithm Definition TLV Sub-TLV Registry	40
18.4.9. Link Attribute Applications Registry	42
19. Acknowledgements	42
20. References	42
20.1. Normative References	42
20.2. Informative References	44
Authors' Addresses	46

1. Introduction

An IGP-computed path based on the shortest IGP metric is often be replaced by a traffic-engineered path due to the traffic requirements which are not reflected by the IGP metric. Some networks engineer the IGP metric assignments in a way that the IGP metric reflects the link bandwidth or delay. If, for example, the IGP metric is reflecting the bandwidth on the link and the application traffic is

delay sensitive, the best IGP path may not reflect the best path from such an application's perspective.

To overcome this limitation, various sorts of traffic engineering have been deployed, including RSVP-TE and SR-TE, in which case the TE component is responsible for computing paths based on additional metrics and/or constraints. Such paths need to be installed in the forwarding tables in addition to, or as a replacement for, the original paths computed by IGPs. Tunnels are often used to represent the engineered paths and mechanisms like one described in [RFC3906] are used to replace the native IGP paths with such tunnel paths.

This document specifies a set of extensions to IS-IS, OSPFv2, and OSPFv3 that enable a router to advertise TLVs that (a) identify calculation-type, (b) specify a metric-type, and (c) describe a set of constraints on the topology, that are to be used to compute the best paths along the constrained topology. A given combination of calculation-type, metric-type, and constraints is known as a "Flexible Algorithm Definition". A router that sends such a set of TLVs also assigns a Flex-Algorithm value to the specified combination of calculation-type, metric-type, and constraints.

This document also specifies a way for a router to use IGPs to associate one or more SR Prefix-SIDs or SRv6 locators with a particular Flex-Algorithm. Each such Prefix-SID or SRv6 locator then represents a path that is computed according to the identified Flex-Algorithm.

2. Requirements Language

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

3. Terminology

This section defines terms that are often used in this document.

Flexible Algorithm Definition (FAD) - the set consisting of (a) calculation-type, (b) metric-type, and (c) a set of constraints.

Flexible Algorithm - a numeric identifier in the range 128-255 that is associated via configuration with the Flexible-Algorithm Definition.

Local Flexible Algorithm Definition - Flexible Algorithm Definition defined locally on the node.

Remote Flexible Algorithm Definition - Flexible Algorithm Definition received from other nodes via IGP flooding.

Flexible Algorithm Participation - per application configuration state that expresses whether the node is participating in a particular Flexible Algorithm.

IGP Algorithm - value from the the "IGP Algorithm Types" registry defined under "Interior Gateway Protocol (IGP) Parameters" IANA registries. IGP Algorithms represents the triplet (Calculation Type, Metric, Constraints), where the second and third elements of the triple MAY be unspecified.

ABR - Area Border Router. In IS-IS terminology it is also known as L1/L2 router.

ASBR - Autonomous System Border Router.

4. Flexible Algorithm

Many possible constraints may be used to compute a path over a network. Some networks are deployed as multiple planes. A simple form of constraint may be to use a particular plane. A more sophisticated form of constraint can include some extended metric as described in [RFC8570]. Constraints which restrict paths to links with specific affinities or avoid links with specific affinities are also possible. Combinations of these are also possible.

To provide maximum flexibility, we want to provide a mechanism that allows a router to (a) identify a particular calculation-type, (b) metric-type, (c) describe a particular set of constraints, and (d) assign a numeric identifier, referred to as Flex-Algorithm, to the combination of that calculation-type, metric-type, and those constraints. We want the mapping between the Flex-Algorithm and its meaning to be flexible and defined by the user. As long as all routers in the domain have a common understanding as to what a particular Flex-Algorithm represents, the resulting routing computation is consistent and traffic is not subject to any looping.

The set consisting of (a) calculation-type, (b) metric-type, and (c) a set of constraints is referred to as a Flexible-Algorithm Definition.

Flexible-Algorithm is a numeric identifier in the range 128-255 that is associated via configuratin with the Flexible-Algorithm Definition.

IANA "IGP Algorithm Types" registry defines the set of values for IGP Algorithms. We propose to allocate the following values for Flex-Algorithms from this registry:

128-255 - Flex-Algorithms

5. Flexible Algorithm Definition Advertisement

To guarantee the loop-free forwarding for paths computed for a particular Flex-Algorithm, all routers that (a) are configured to participate in a particular Flex-Algorithm, and (b) are in the same Flex-Algorithm definition advertisement scope MUST agree on the definition of the Flex-Algorithm.

5.1. IS-IS Flexible Algorithm Definition Sub-TLV

The IS-IS Flexible Algorithm Definition Sub-TLV (FAD Sub-TLV) is used to advertise the definition of the Flex-Algorithm.

The IS-IS FAD Sub-TLV is advertised as a Sub-TLV of the IS-IS Router Capability TLV-242 that is defined in [RFC7981].

IS-IS FAD Sub-TLV has the following format:

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
+	+	+	+
	Calc-Type		Priority
+	+	+	+
	Sub-TLVs		
+			+
	...		
+			+
+	+	+	+

where:

Type: 26

Length: variable, dependent on the included Sub-TLVs

Flex-Algorithm: Single octet value between 128 and 255 inclusive.

Metric-Type: Type of metric to be used during the calculation.
Following values are defined:

0: IGP Metric

1: Min Unidirectional Link Delay as defined in [RFC8570], section 4.2, encoded as application specific link attribute as specified in [RFC8919] and Section 12 of this document.

2: Traffic Engineering Default Metric as defined in [RFC5305], section 3.7, encoded as application specific link attribute as specified in [RFC8919] and Section 12 of this document.

Calc-Type: value from 0 to 127 inclusive from the "IGP Algorithm Types" registry defined under "Interior Gateway Protocol (IGP) Parameters" IANA registries. IGP algorithms in the range of 0-127 have a defined triplet (Calculation Type, Metric, Constraints). When used to specify the Calc-Type in the FAD Sub-TLV, only the Calculation Type defined for the specified IGP Algorithm is used. The Metric/Constraints MUST NOT be inherited. If the required calculation type is Shortest Path First, the value 0 SHOULD appear in this field.

Priority: Value between 0 and 255 inclusive that specifies the priority of the advertisement.

Sub-TLVs - optional sub-TLVs.

The IS-IS FAD Sub-TLV MAY be advertised in an LSP of any number. IS-IS router MAY advertise more than one IS-IS FAD Sub-TLV for a given Flexible-Algorithm (see Section 6).

The IS-IS FAD Sub-TLV has an area scope. The Router Capability TLV in which the FAD Sub-TLV is present MUST have the S-bit clear.

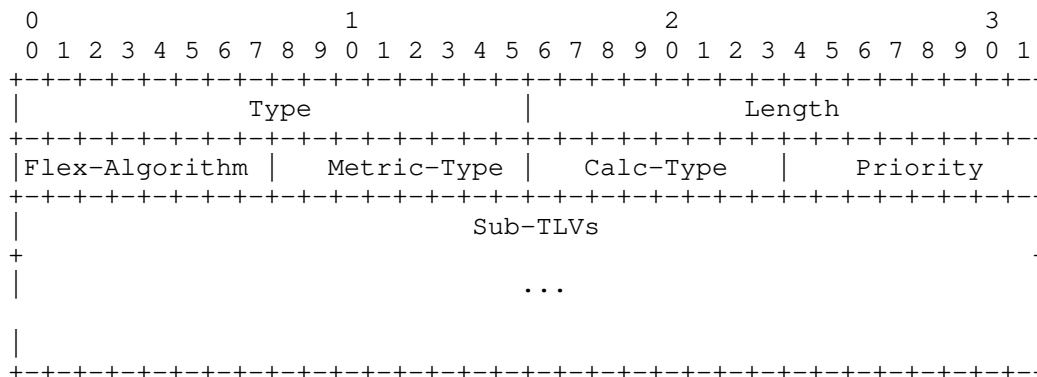
IS-IS L1/L2 router MAY be configured to re-generate the winning FAD from level 2, without any modification to it, to level 1 area. The re-generation of the FAD Sub-TLV from level 2 to level 1 is determined by the L1/L2 router, not by the originator of the FAD advertisement in the level 2. In such case, the re-generated FAD Sub-TLV will be advertised in the level 1 Router Capability TLV originated by the L1/L2 router.

L1/L2 router MUST NOT re-generate any FAD Sub-TLV from level 1 to level 2.

5.2. OSPF Flexible Algorithm Definition TLV

OSPF FAD TLV is advertised as a top-level TLV of the RI LSA that is defined in [RFC7770].

OSPF FAD TLV has the following format:



where:

Type: 16

Length: variable, dependent on the included Sub-TLVs

Flex-Algorithm:: Flex-Algorithm number. Value between 128 and 255 inclusive.

Metric-Type: Type of metric to be used during the calculation.
Following values are defined:

0: IGP Metric

1: Min Unidirectional Link Delay as defined in [RFC7471], section 4.2, encoded as application specific link attribute as specified in [RFC8920] and Section 12 of this document.

2: Traffic Engineering metric as defined in [RFC3630], section 2.5.5, encoded as application specific link attribute as specified in [RFC8920] and Section 12 of this document.

Calc-Type: as described in Section 5.1

Priority: as described in Section 5.1

Sub-TLVs - optional sub-TLVs.

When multiple OSPF FAD TLVs, for the same Flexible-Algorithm, are received from a given router, the receiver MUST use the first occurrence of the TLV in the Router Information LSA. If the OSPF FAD TLV, for the same Flex-Algorithm, appears in multiple Router Information LSAs that have different flooding scopes, the OSPF FAD TLV in the Router Information LSA with the area-scoped flooding scope MUST be used. If the OSPF FAD TLV, for the same algorithm, appears in multiple Router Information LSAs that have the same flooding scope, the OSPF FAD TLV in the Router Information (RI) LSA with the numerically smallest Instance ID MUST be used and subsequent instances of the OSPF FAD TLV MUST be ignored.

The RI LSA can be advertised at any of the defined opaque flooding scopes (link, area, or Autonomous System (AS)). For the purpose of OSPF FAD TLV advertisement, area-scoped flooding is REQUIRED. The Autonomous System flooding scope SHOULD NOT be used by default unless local configuration policy on the originating router indicates domain wide flooding.

5.3. Common Handling of Flexible Algorithm Definition TLV

This section describes the protocol-independent handling of the FAD TLV (OSPF) or FAD Sub-TLV (IS-IS). We will refer to it as FAD TLV in this section, even though in the case of IS-IS it is a Sub-TLV.

The value of the Flex-Algorithm MUST be between 128 and 255 inclusive. If it is not, the FAD TLV MUST be ignored.

Only a subset of the routers participating in the particular Flex-Algorithm need to advertise the definition of the Flex-Algorithm.

Every router, that is configured to participate in a particular Flex-Algorithm, MUST select the Flex-Algorithm definition based on the following ordered rules. This allows for the consistent Flex-Algorithm definition selection in cases where different routers advertise different definitions for a given Flex-Algorithm:

1. From the advertisements of the FAD in the area (including both locally generated advertisements and received advertisements) select the one(s) with the highest priority value.
2. If there are multiple advertisements of the FAD with the same highest priority, select the one that is originated from the router with the highest System-ID, in the case of IS-IS, or Router ID, in the case of OSPFv2 and OSPFv3. For IS-IS, the System-ID is

described in [ISO10589]. For OSPFv2 and OSPFv3, standard Router ID is described in [RFC2328] and [RFC5340] respectively.

A router that is not configured to participate in a particular Flex-Algorithm MUST ignore FAD Sub-TLVs advertisements for such Flex-Algorithm.

A router that is not participating in a particular Flex-Algorithm is allowed to advertise FAD for such Flex-Algorithm. Receiving routers MUST consider FAD advertisement regardless of the Flex-Algorithm participation of the FAD originator.

Any change in the Flex-Algorithm definition may result in temporary disruption of traffic that is forwarded based on such Flex-Algorithm paths. The impact is similar to any other event that requires network-wide convergence.

If a node is configured to participate in a particular Flexible-Algorithm, but there is no valid Flex-Algorithm definition available for it, or the selected Flex-Algorithm definition includes calculation-type, metric-type, constraint, flag, or Sub-TLV that is not supported by the node, it MUST stop participating in such Flexible-Algorithm. That implies that it MUST NOT announce participation for such Flexible-Algorithm as specified in Section 11 and it MUST remove any forwarding state associated with it.

Flex-Algorithm definition is topology independent. It applies to all topologies that a router participates in.

6. Sub-TLVs of IS-IS FAD Sub-TLV

One of the limitations of IS-IS [ISO10589] is that the length of a TLV/sub-TLV is limited to a maximum of 255 octets. For the FAD sub-TLV, there are a number of sub-sub-TLVs (defined below) which are supported. For a given Flex-Algorithm, it is possible that the total number of octets required to completely define a FAD exceeds the maximum length supported by a single FAD sub-TLV. In such cases, the FAD may be split into multiple such sub-TLVs and the content of the multiple FAD sub-TLVs combined to provide a complete FAD for the Flex-Algorithm. In such case, the fixed portion of the FAD (see Section 5.1) MUST be identical in all FAD sub-TLVs for a given Flex-Algorithm from a given IS. In case the fixed portion of such FAD Sub-TLVs differ, the values in the fixed portion in the FAD sub-TLV in the first occurrence in the lowest numbered LSP from a given IS MUST be used.

Any specification that introduces a new ISIS FAD sub-sub-TLV MUST specify whether the FAD sub-TLV may appear multiple times in the set

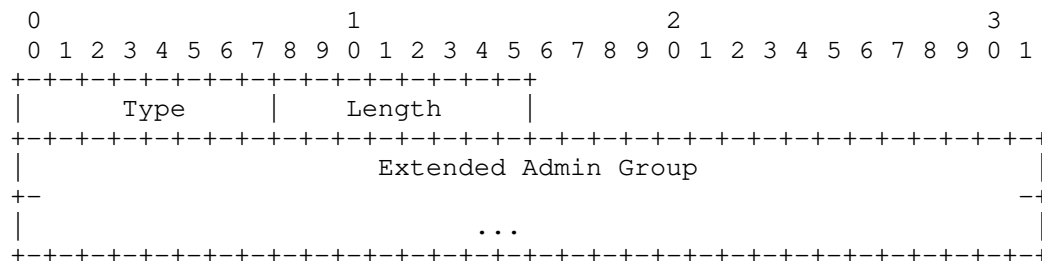
of FAD sub-TLVs for a given Flex-Algorithm from a given IS and how to handle them if multiple are allowed.

6.1. IS-IS Flexible Algorithm Exclude Admin Group Sub-TLV

The Flexible Algorithm definition can specify 'colors' that are used by the operator to exclude links during the Flex-Algorithm path computation.

The IS-IS Flexible Algorithm Exclude Admin Group Sub-TLV is used to advertise the exclude rule that is used during the Flex-Algorithm path calculation as specified in Section 13.

The IS-IS Flexible Algorithm Exclude Admin Group Sub-TLV (FAEAG Sub-TLV) is a Sub-TLV of the IS-IS FAD Sub-TLV. It has the following format:



where:

Type: 1

Length: variable, dependent on the size of the Extended Admin Group. MUST be a multiple of 4 octets.

Extended Administrative Group: Extended Administrative Group as defined in [RFC7308].

The IS-IS FAEAG Sub-TLV MUST NOT appear more than once in a single IS-IS FAD Sub-TLV. If it appears more than once, the IS-IS FAD Sub-TLV MUST be ignored by the receiver.

The IS-IS FAEAG Sub-TLV MUST NOT appear more than once in the set of FAD sub-TLVs for a given Flex-Algorithm from a given IS. If it appears more than once in such set, the IS-IS FAEAG Sub-TLV in the first occurrence in the lowest numbered LSP from a given IS MUST be used and any other occurrences MUST be ignored.

6.2. IS-IS Flexible Algorithm Include-Any Admin Group Sub-TLV

The Flexible Algorithm definition can specify 'colors' that are used by the operator to include links during the Flex-Algorithm path computation.

The IS-IS Flexible Algorithm Include-Any Admin Group Sub-TLV is used to advertise include-any rule that is used during the Flex-Algorithm path calculation as specified in Section 13.

The format of the IS-IS Flexible Algorithm Include-Any Admin Group Sub-TLV is identical to the format of the FAEAG Sub-TLV in Section 6.1.

The IS-IS Flexible Algorithm Include-Any Admin Group Sub-TLV Type is 2.

The IS-IS Flexible Algorithm Include-Any Admin Group Sub-TLV MUST NOT appear more than once in a single IS-IS FAD Sub-TLV. If it appears more than once, the IS-IS FAD Sub-TLV MUST be ignored by the receiver.

The IS-IS Flexible Algorithm Include-Any Admin Group Sub-TLV MUST NOT appear more than once in the set of FAD sub-TLVs for a given Flex-Algorithm from a given IS. If it appears more than once in such set, the IS-IS Flexible Algorithm Include-Any Admin Group Sub-TLV in the first occurrence in the lowest numbered LSP from a given IS MUST be used and any other occurrences MUST be ignored.

6.3. IS-IS Flexible Algorithm Include-All Admin Group Sub-TLV

The Flexible Algorithm definition can specify 'colors' that are used by the operator to include link during the Flex-Algorithm path computation.

The IS-IS Flexible Algorithm Include-All Admin Group Sub-TLV is used to advertise include-all rule that is used during the Flex-Algorithm path calculation as specified in Section 13.

The format of the IS-IS Flexible Algorithm Include-All Admin Group Sub-TLV is identical to the format of the FAEAG Sub-TLV in Section 6.1.

The IS-IS Flexible Algorithm Include-All Admin Group Sub-TLV Type is 3.

The IS-IS Flexible Algorithm Include-All Admin Group Sub-TLV MUST NOT appear more than once in a single IS-IS FAD Sub-TLV. If it appears

more than once, the IS-IS FAD Sub-TLV MUST be ignored by the receiver.

The IS-IS Flexible Algorithm Include-All Admin Group Sub-TLV MUST NOT appear more than once in the set of FAD sub-TLVs for a given Flex-Algorithm from a given IS. If it appears more than once in such set, the IS-IS Flexible Algorithm Include-All Admin Group Sub-TLV in the first occurrence in the lowest numbered LSP from a given IS MUST be used and any other occurrences MUST be ignored.

6.4. IS-IS Flexible Algorithm Definition Flags Sub-TLV

The IS-IS Flexible Algorithm Definition Flags Sub-TLV (FADF Sub-TLV) is a Sub-TLV of the IS-IS FAD Sub-TLV. It has the following format:

```

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

```

where:

Type: 4

Length: variable, non-zero number of octets of the Flags field

Flags:

```

      0 1 2 3 4 5 6 7...
+---+---+---+---+---+---+---+
|M| | | | | | |...
+---+---+---+---+---+---+---+

```

M-flag: when set, the Flex-Algorithm specific prefix metric MUST be used for inter-area and external prefix calculation. This flag is not applicable to prefixes advertised as SRv6 locators.

Bits are defined/sent starting with Bit 0 defined above. Additional bit definitions that may be defined in the future SHOULD be assigned in ascending bit order so as to minimize the number of bits that will need to be transmitted.

Undefined bits MUST be transmitted as 0.

Bits that are NOT transmitted MUST be treated as if they are set to 0 on receipt.

The IS-IS FADF Sub-TLV MUST NOT appear more than once in a single IS-IS FAD Sub-TLV. If it appears more than once, the IS-IS FAD Sub-TLV MUST be ignored by the receiver.

The IS-IS FADF Sub-TLV MUST NOT appear more than once in the set of FAD sub-TLVs for a given Flex-Algorithm from a given IS. If it appears more than once in such set, the IS-IS FADF Sub-TLV in the first occurrence in the lowest numbered LSP from a given IS MUST be used and any other occurrences MUST be ignored.

If the IS-IS FADF Sub-TLV is not present inside the IS-IS FAD Sub-TLV, all the bits are assumed to be set to 0.

If a node is configured to participate in a particular Flexible-Algorithm, but the selected Flex-Algorithm definition includes a bit in the IS-IS FADF Sub-TLV that is not supported by the node, it MUST stop participating in such Flexible-Algorithm.

New flag bits may be defined in the future. Implementations MUST check all advertised flag bits in the received IS-IS FADF Sub-TLV - not just the subset currently defined.

6.5. IS-IS Flexible Algorithm Exclude SRLG Sub-TLV

The Flexible Algorithm definition can specify Shared Risk Link Groups (SRLGs) that the operator wants to exclude during the Flex-Algorithm path computation.

The IS-IS Flexible Algorithm Exclude SRLG Sub-TLV (FAESRLG) is used to advertise the exclude rule that is used during the Flex-Algorithm path calculation as specified in Section 13.

The IS-IS FAESRLG Sub-TLV is a Sub-TLV of the IS-IS FAD Sub-TLV. It has the following format:

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|          Type          |      Length      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               Shared Risk Link Group Value                               |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                                                                               ...                               |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

where:

Type: 5

Length: variable, dependent on number of SRLG values. MUST be a multiple of 4 octets.

Shared Risk Link Group Value: SRLG value as defined in [RFC5307].

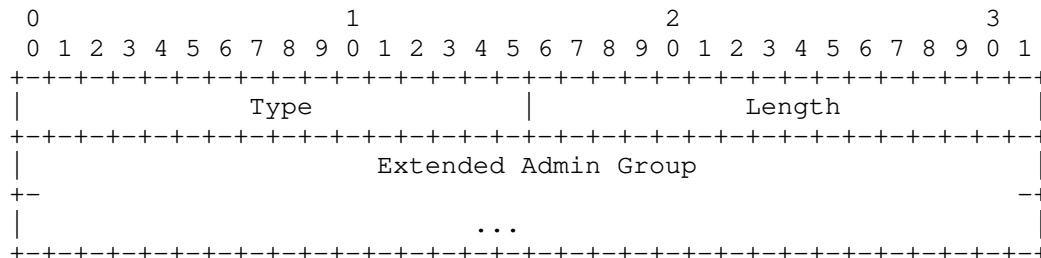
The IS-IS FAESRLG Sub-TLV MUST NOT appear more than once in a single IS-IS FAD Sub-TLV. If it appears more than once, the IS-IS FAD Sub-TLV MUST be ignored by the receiver.

The IS-IS FAESRLG Sub-TLV MAY appear more than once in the set of FAD sub-TLVs for a given Flex-Algorithm from a given IS. This may be necessary in cases where the total number of SRLG values which are specified cause the FAD sub-TLV to exceed the maximum length of a single FAD sub-TLV. In such case the receiver MUST use the union of all values across all IS-IS FAESRLG Sub-TLVs from such set.

7. Sub-TLVs of OSPF FAD TLV

7.1. OSPF Flexible Algorithm Exclude Admin Group Sub-TLV

The Flexible Algorithm Exclude Admin Group Sub-TLV (FAEAG Sub-TLV) is a Sub-TLV of the OSPF FAD TLV. It's usage is described in Section 6.1. It has the following format:



where:

Type: 1

Length: variable, dependent on the size of the Extended Admin Group. MUST be a multiple of 4 octets.

Extended Administrative Group: Extended Administrative Group as defined in [RFC7308].

The OSPF FAEAG Sub-TLV MUST NOT appear more than once in an OSPF FAD TLV. If it appears more than once, the OSPF FAD TLV MUST be ignored by the receiver.

7.2. OSPF Flexible Algorithm Include-Any Admin Group Sub-TLV

The usage of this Sub-TLVs is described in Section 6.2.

The format of the OSPF Flexible Algorithm Include-Any Admin Group Sub-TLV is identical to the format of the OSPF FAEAG Sub-TLV in Section 7.1.

The OSPF Flexible Algorithm Include-Any Admin Group Sub-TLV Type is 2.

The OSPF Flexible Algorithm Include-Any Admin Group Sub-TLV MUST NOT appear more than once in an OSPF FAD TLV. If it appears more than once, the OSPF FAD TLV MUST be ignored by the receiver.

7.3. OSPF Flexible Algorithm Include-All Admin Group Sub-TLV

The usage of this Sub-TLVs is described in Section 6.3.

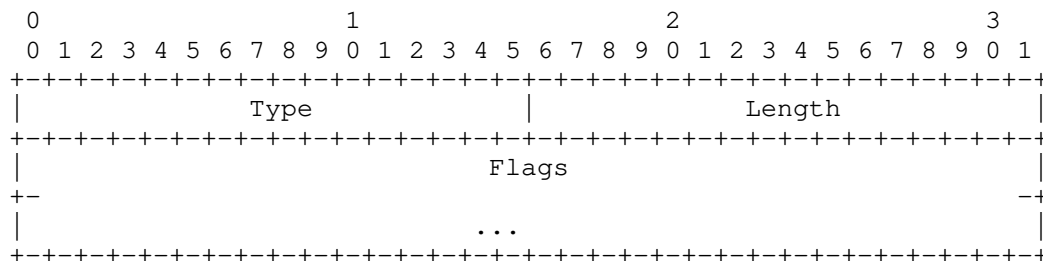
The format of the OSPF Flexible Algorithm Include-All Admin Group Sub-TLV is identical to the format of the OSPF FAEAG Sub-TLV in Section 7.1.

The OSPF Flexible Algorithm Include-All Admin Group Sub-TLV Type is 3.

The OSPF Flexible Algorithm Include-All Admin Group Sub-TLV MUST NOT appear more than once in an OSPF FAD TLV. If it appears more than once, the OSPF FAD TLV MUST be ignored by the receiver.

7.4. OSPF Flexible Algorithm Definition Flags Sub-TLV

The OSPF Flexible Algorithm Definition Flags Sub-TLV (FADF Sub-TLV) is a Sub-TLV of the OSPF FAD TLV. It has the following format:



where:

Type: 4

Length: variable, dependent on the size of the Flags field. MUST be a multiple of 4 octets.

Flags:

```

    0 1 2 3 4 5 6 7...
    +-+-+-+-+-+-+-+...
    |M| | |         ...
    +-+-+-+-+-+-+-+...

```

M-flag: when set, the Flex-Algorithm specific prefix and ASBR metric MUST be used for inter-area and external prefix calculation. This flag is not applicable to prefixes advertised as SRv6 locators.

Bits are defined/sent starting with Bit 0 defined above. Additional bit definitions that may be defined in the future SHOULD be assigned in ascending bit order so as to minimize the number of bits that will need to be transmitted.

Undefined bits MUST be transmitted as 0.

Bits that are NOT transmitted MUST be treated as if they are set to 0 on receipt.

The OSPF FADF Sub-TLV MUST NOT appear more than once in an OSPF FAD TLV. If it appears more than once, the OSPF FAD TLV MUST be ignored by the receiver.

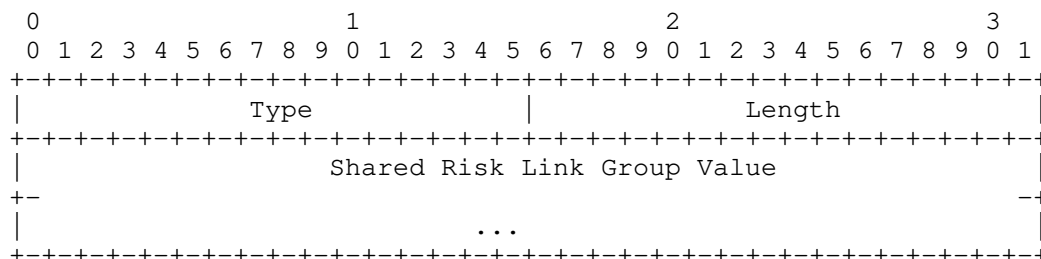
If the OSPF FADF Sub-TLV is not present inside the OSPF FAD TLV, all the bits are assumed to be set to 0.

If a node is configured to participate in a particular Flexible-Algorithm, but the selected Flex-Algorithm definition includes a bit in the OSPF FADF Sub-TLV that is not supported by the node, it MUST stop participating in such Flexible-Algorithm.

New flag bits may be defined in the future. Implementations MUST check all advertised flag bits in the received OSPF FADF Sub-TLV - not just the subset currently defined.

7.5. OSPF Flexible Algorithm Exclude SRLG Sub-TLV

The OSPF Flexible Algorithm Exclude SRLG Sub-TLV (FAESRLG Sub-TLV) is a Sub-TLV of the OSPF FAD TLV. Its usage is described in Section 6.5. It has the following format:



where:

Type: 5

Length: variable, dependent on the number of SRLGs. MUST be a multiple of 4 octets.

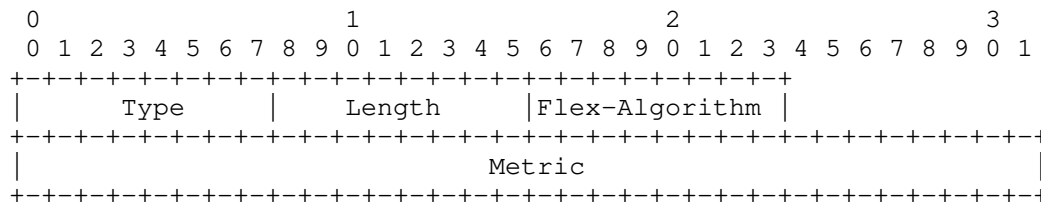
Shared Risk Link Group Value: SRLG value as defined in [RFC4203].

The OSPF FAESRLG Sub-TLV MUST NOT appear more than once in an OSPF FAD TLV. If it appears more than once, the OSPF FAD TLV MUST be ignored by the receiver.

8. IS-IS Flexible Algorithm Prefix Metric Sub-TLV

The IS-IS Flexible Algorithm Prefix Metric (FAPM) Sub-TLV supports the advertisement of a Flex-Algorithm specific prefix metric associated with a given prefix advertisement.

The IS-IS FAPM Sub-TLV is a sub-TLV of TLVs 135, 235, 236, and 237 and has the following format:



where:

Type: 6

Length: 5 octets

Flex-Algorithm: Single octet value between 128 and 255 inclusive.

Metric: 4 octets of metric information

The IS-IS FAPM Sub-TLV MAY appear multiple times in its parent TLV. If it appears more than once with the same Flex-Algorithm value, the first instance MUST be used and any subsequent instances MUST be ignored.

If a prefix is advertised with a Flex-Algorithm prefix metric larger than MAX_PATH_METRIC as defined in [RFC5305] this prefix MUST NOT be considered during the Flexible-Algorithm computation.

The usage of the Flex-Algorithm prefix metric is described in Section 13.

The IS-IS FAPM Sub-TLV MUST NOT be advertised as a sub-TLV of the IS-IS SRv6 Locator TLV [I-D.ietf-lsr-isis-srv6-extensions]. The IS-IS SRv6 Locator TLV includes the Algorithm and Metric fields which MUST be used instead. If the FAPM Sub-TLV is present as a sub-TLV of the IS-IS SRv6 Locator TLV in the received LSP, such FAPM Sub-TLV MUST be ignored.

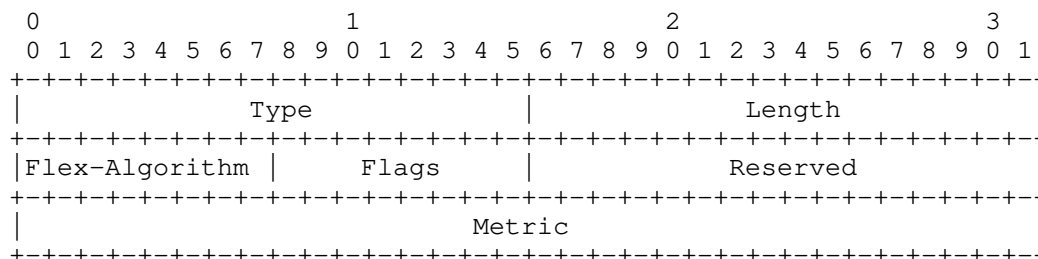
9. OSPF Flexible Algorithm Prefix Metric Sub-TLV

The OSPF Flexible Algorithm Prefix Metric (FAPM) Sub-TLV supports the advertisement of a Flex-Algorithm specific prefix metric associated with a given prefix advertisement.

The OSPF Flex-Algorithm Prefix Metric (FAPM) Sub-TLV is a Sub-TLV of the:

- OSPFv2 Extended Prefix TLV [RFC7684]
- Following OSPFv3 TLVs as defined in [RFC8362]:
 - Inter-Area Prefix TLV
 - External Prefix TLV

OSPF FAPM Sub-TLV has the following format:



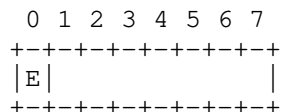
where:

Type: 3 for OSPFv2, 26 for OSPFv3

Length: 8 octets

Flex-Algorithm: Single octet value between 128 and 255 inclusive.

Flags: single octet value



E bit : position 0: The type of external metric. If bit is set, the metric specified is a Type 2 external metric. This bit is applicable only to OSPF External and NSSA external prefixes. This is semantically the same as E bit in section A.4.5 of [RFC2328] and section A.4.7 of [RFC5340] for OSPFv2 and OSPFv3 respectively.

Bits 1 through 7: MUST be cleared by sender and ignored by receiver.

Reserved: Must be set to 0, ignored at reception.

Metric: 4 octets of metric information

The OSPF FAPM Sub-TLV MAY appear multiple times in its parent TLV. If it appears more than once with the same Flex-Algorithm value, the first instance MUST be used and any subsequent instances MUST be ignored.

The usage of the Flex-Algorithm prefix metric is described in Section 13.

10. OSPF Flexible Algorithm ASBR Reachability Advertisement

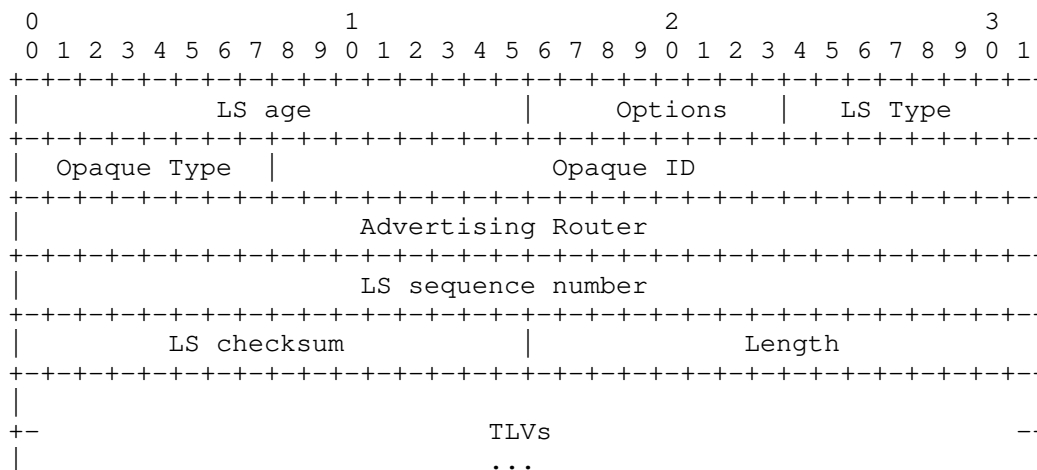
An OSPF ABR advertises the reachability of ASBRs in its attached areas to enable routers within those areas to perform route calculations for external prefixes advertised by the ASBRs. OSPF extensions for advertisement of Flex-Algorithm specific reachability and metric for ASBRs is similarly required for Flex-Algorithm external prefix computations as described further in Section 13.1.

10.1. OSPFv2 Extended Inter-Area ASBR LSA

The OSPFv2 Extended Inter-Area ASBR (EIA-ASBR) LSA is an OSPF Opaque LSA [RFC5250] that is used to advertise additional attributes related to the reachability of the OSPFv2 ASBR that is external to the area yet internal to the OSPF domain. Semantically, the OSPFv2 EIA-ASBR LSA is equivalent to the fixed format Type 4 Summary LSA [RFC2328]. Unlike the Type 4 Summary LSA, the LSID of the EIA-ASBR LSA does not carry the ASBR Router-ID - the ASBR Router-ID is carried in the body of the LSA. OSPFv2 EIA-ASBR LSA is advertised by an OSPFv2 ABR and its flooding is defined to be area-scoped only.

An OSPFv2 ABR generates the EIA-ASBR LSA for an ASBR when it is advertising the Type-4 Summary LSA for it and has the need for advertising additional attributes for that ASBR beyond what is conveyed in the fixed format Type-4 Summary LSA. An OSPFv2 ABR MUST NOT advertise the EIA-ASBR LSA for an ASBR for which it is not advertising the Type 4 Summary LSA. This ensures that the ABR does not generate the EIA-ASBR LSA for an ASBR to which it does not have reachability in the base OSPFv2 topology calculation. The OSPFv2 ABR SHOULD NOT advertise the EIA-ASBR LSA for an ASBR when it does not have additional attributes to advertise for that ASBR.

The OSPFv2 EIA-ASBR LSA has the following format:



The Opaque Type used by the OSPFv2 EIA-ASBR LSA is TBD (suggested value 11). The Opaque Type is used to differentiate the various types of OSPFv2 Opaque LSAs and is described in Section 3 of [RFC5250]. The LS Type MUST be 10, indicating that the Opaque LSA flooding scope is area-local [RFC5250]. The LSA Length field [RFC2328] represents the total length (in octets) of the Opaque LSA, including the LSA header and all TLVs (including padding).

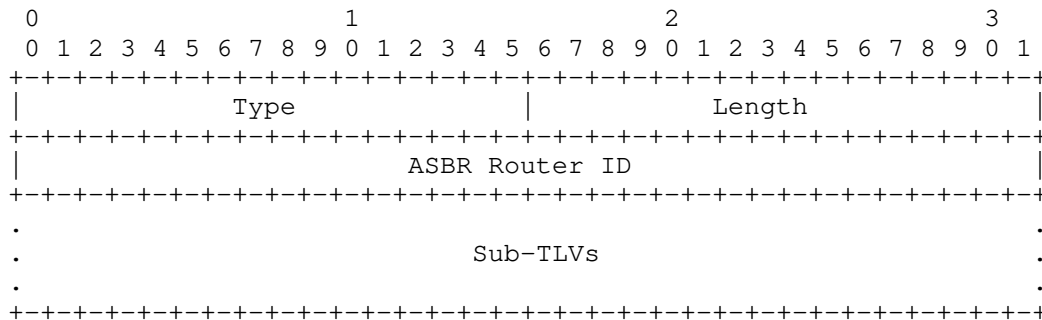
The Opaque ID field is an arbitrary value used to maintain multiple OSPFv2 EIA-ASBR LSAs. For OSPFv2 EIA-ASBR LSAs, the Opaque ID has no semantic significance other than to differentiate OSPFv2 EIA-ASBR LSAs originated by the same OSPFv2 ABR. If multiple OSPFv2 EIA-ASBR LSAs specify the same ASBR, the attributes from the Opaque LSA with the lowest Opaque ID SHOULD be used.

The format of the TLVs within the body of the OSPFv2 EIA-ASBR LSA is the same as the format used by the Traffic Engineering Extensions to OSPFv2 [RFC3630]. The variable TLV section consists of one or more nested TLV tuples. Nested TLVs are also referred to as sub-TLVs. The Length field defines the length of the value portion in octets (thus, a TLV with no value portion would have a length of 0). The TLV is padded to 4-octet alignment; padding is not included in the Length field (so a 3-octet value would have a length of 3, but the total size of the TLV would be 8 octets). Nested TLVs are also 32-bit aligned. For example, a 1-byte value would have the Length field set to 1, and 3 octets of padding would be added to the end of the value portion of the TLV. The padding is composed of zeros.

10.1.1. OSPFv2 Extended Inter-Area ASBR TLV

The OSPFv2 Extended Inter-Area ASBR (EIA-ASBR) TLV is a top-level TLV of the OSPFv2 EIA-ASBR LSA and is used to advertise additional attributes associated with the reachability of an ASBR.

The OSPFv2 EIA-ASBR TLV has the following format:



where:

Type: 1

Length: variable

ASBR Router ID: four octets carrying the OSPF Router ID of the ASBR whose information is being carried.

Sub-TLVs : variable

Only a single OSPFv2 EIA-ASBR TLV MUST be advertised in each OSPFv2 EIA-ASBR LSA and the receiver MUST ignore all instances of this TLV other than the first one in an LSA.

OSPFv2 EIA-ASBR TLV MUST be present inside an OSPFv2 EIA-ASBR LSA with at least a single sub-TLV included, otherwise the OSPFv2 EIA-ASBR LSA MUST be ignored by the receiver.

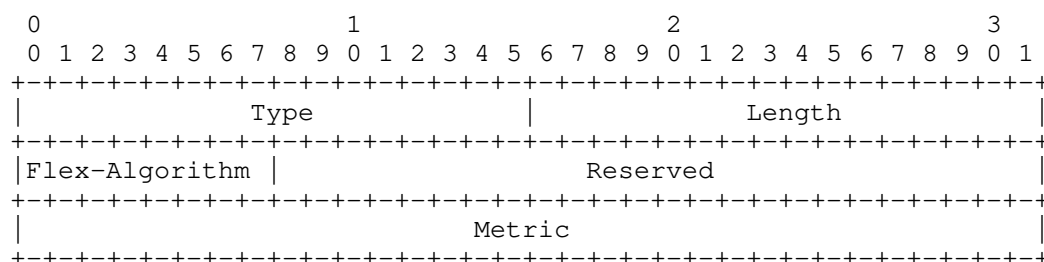
10.2. OSPF Flexible Algorithm ASBR Metric Sub-TLV

The OSPF Flexible Algorithm ASBR Metric (FAAM) Sub-TLV supports the advertisement of a Flex-Algorithm specific metric associated with a given ASBR reachability advertisement by an ABR.

The OSPF Flex-Algorithm ASBR Metric (FAAM) Sub-TLV is a Sub-TLV of the:

- OSPFv2 Extended Inter-Area ASBR TLV as defined in Section 10.1.1
- OSPFv3 Inter-Area-Router TLV defined in [RFC8362]

OSPF FAAM Sub-TLV has the following format:



where:

Type: 1 for OSPFv2, TBD (suggested value 30) for OSPFv3

Length: 8 octets

Flex-Algorithm: Single octet value between 128 and 255 inclusive.

Reserved: Must be set to 0, ignored at reception.

Metric: 4 octets of metric information

The OSPF FAAM Sub-TLV MAY appear multiple times in its parent TLV. If it appears more than once with the same Flex-Algorithm value, the first instance MUST be used and any subsequent instances MUST be ignored.

The advertisement of the ASBR reachability using the OSPF FAAM Sub-TLV inside the OSPFv2 EIA-ASBR LSA follows the section 12.4.3 of [RFC2328] and inside the OSPFv3 E-Inter-Area-Router LSA follows the section 4.8.5 of [RFC5340]. The reachability of the ASBR is evaluated in the context of the specific Flex-Algorithm.

The FAAM computed by the ABR will be equal to the metric to reach the ASBR for a given Flex-Algorithm in a source area or the cumulative metric via other ABR(s) when the ASBR is in a remote area. This is similar in nature to how the metric is set when the ASBR reachability metric is computed in the default algorithm for the metric in the OSPFv2 Type 4 ASBR Summary LSA and the OSPFv3 Inter-Area-Router LSA.

An OSPF ABR MUST NOT include the OSPF FAAM Sub-TLV with a specific Flex-Algorithm in its reachability advertisement for an ASBR between

areas unless that ASBR is reachable for it in the context of that specific Flex-Algorithm.

An OSPF ABR MUST include the OSPF FAAM Sub-TLVs as part of the ASBR reachability advertisement between areas for the Flex-Algorithm for which the winning FAD includes the M-flag and the ASBR is reachable in the context of that specific Flex-Algorithm.

OSPF routers MUST use the OSPF FAAM Sub-TLV to calculate the reachability of the ASBRs if the winning FAD for the specific Flex-Algorithm includes the M-flag. OSPF routers MUST NOT use the OSPF FAAM Sub-TLV to calculate the reachability of the ASBRs for the specific Flex-Algorithm if the winning FAD for such Flex-Algorithm does not include the M-flag. Instead, the OSPFv2 Type 4 Summary LSAs or the OSPFv3 Inter-Area-Router-LSAs MUST be used instead as specified in section 16.2 of [RFC2328] and section 4.8.5 of [RFC5340] for OSPFv2 and OSPFv3 respectively.

The processing of the new or changed OSPF FAAM Sub-TLV triggers the processing of the External routes similar to what is described in section 16.5 of the [RFC2328] for OSPFv2 and section 4.8.5 of [RFC5340] for OSPFv3 for the specific Flex-Algorithm. The External and NSSA External route calculation should be limited to Flex-Algorithm(s) for which the winning FAD(s) includes the M-flag.

Processing of the OSPF FAAM Sub-TLV does not require the existence of the equivalent OSPFv2 Type 4 Summary LSA or the OSPFv3 Inter-Area-Router-LSA that is advertised by the same ABR inside the area. When the OSPFv2 EIA-ASBR LSA or the OSPFv3 E-Inter-Area-Router-LSA are advertised along with the OSPF FAAM Sub-TLV by the ABR for a specific ASBR, it is expected that the same ABR would advertise the reachability of the same ASBR in the equivalent base LSAs - i.e., the OSPFv2 Type 4 Summary LSA or the OSPFv3 Inter-Area-Router-LSA. The presence of the base LSA is not mandatory for the usage of the extended LSA with the OSPF FAAM Sub-TLV. This means that the order in which these LSAs are received is not significant.

11. Advertisement of Node Participation in a Flex-Algorithm

When a router is configured to support a particular Flex-Algorithm, we say it is participating in that Flex-Algorithm.

Paths computed for a specific Flex-Algorithm MAY be used by various applications, each potentially using its own specific data plane for forwarding traffic over such paths. To guarantee the presence of the application specific forwarding state associated with a particular Flex-Algorithm, a router MUST advertise its participation for a particular Flex-Algorithm for each application specifically.

11.1. Advertisement of Node Participation for Segment Routing

[RFC8667], [RFC8665], and [RFC8666] (IGP Segment Routing extensions) describe how the SR-Algorithm is used to compute the IGP best path.

Routers advertise the support for the SR-Algorithm as a node capability as described in the above mentioned IGP Segment Routing extensions. To advertise participation for a particular Flex-Algorithm for Segment Routing, including both SR MPLS and SRv6, the Flex-Algorithm value MUST be advertised in the SR-Algorithm TLV (OSPF) or sub-TLV (IS-IS).

Segment Routing Flex-Algorithm participation advertisement is topology independent. When a router advertises participation in an SR-Algorithm, the participation applies to all topologies in which the advertising node participates.

11.2. Advertisement of Node Participation for Other Applications

This section describes considerations related to how other applications can advertise their participation in a specific Flex-Algorithm.

Application-specific Flex-Algorithm participation advertisements MAY be topology specific or MAY be topology independent, depending on the application itself.

Application-specific advertisement for Flex-Algorithm participation MUST be defined for each application and is outside of the scope of this document.

12. Advertisement of Link Attributes for Flex-Algorithm

Various link attributes may be used during the Flex-Algorithm path calculation. For example, include or exclude rules based on link affinities can be part of the Flex-Algorithm definition as defined in Section 6 and Section 7.

Application-specific link attributes, as specified in [RFC8919] or [RFC8920], that are to be used during Flex-Algorithm calculation MUST use the Application-Specific Link Attribute (ASLA) advertisements defined in [RFC8919] or [RFC8920], unless, in the case of IS-IS, the L-Flag is set in the ASLA advertisement. When the L-Flag is set, then legacy advertisements are to be used, subject to the procedures and constraints defined in [[RFC8919] Section 4.2 and Section 6.

The mandatory use of ASLA advertisements applies to link attributes specifically mentioned in this document (Min Unidirectional Link

Delay, TE Default Metric, Administrative Group, Extended Administrative Group and Shared Risk Link Group) and any other link attributes that may be used in support of Flex-Algorithm in the future.

A new Application Identifier Bit is defined to indicate that the ASLA advertisement is associated with the Flex-Algorithm application. This bit is set in the Standard Application Bit Mask (SABM) defined in [RFC8919] or [RFC8920]:

Bit-3: Flexible Algorithm (X-bit)

ASLA Admin Group Advertisements to be used by the Flexible Algorithm Application MAY use either the Administrative Group or Extended Administrative Group encodings. If the Administrative Group encoding is used, then the first 32 bits of the corresponding FAD sub-TLVs are mapped to the link attribute advertisements as specified in RFC 7308.

A receiver supporting this specification MUST accept both ASLA Administrative Group and Extended Administrative Group TLVs as defined in [RFC8919] or [RFC8920]. In the case of ISIS, if the L-Flag is set in ASLA advertisement, as defined in [RFC8919] Section 4.2, then the receiver MUST be able to accept both Administrative Group TLV as defined in [RFC5305] and Extended Administrative Group TLV as defined in [RFC7308].

13. Calculation of Flexible Algorithm Paths

A router MUST be configured to participate in a given Flex-Algorithm K and MUST select the FAD based on the rules defined in Section 5.3 before it can compute any path for that Flex-Algorithm.

No specific two way connectivity check is performed during the Flex-Algorithm path computation. The result of the existing, Flex-Algorithm agnostic, two way connectivity check is used during the Flex-Algorithm path computation.

As described in Section 11, participation for any particular Flex-Algorithm MUST be advertised on a per-application basis. Calculation of the paths for any particular Flex-Algorithm MUST be application specific.

The way applications handle nodes that do not participate in Flexible-Algorithm is application specific. If the application only wants to consider participating nodes during the Flex-Algorithm calculation, then when computing paths for a given Flex-Algorithm, all nodes that do not advertise participation for that Flex-Algorithm in their application-specific advertisements MUST be pruned from the

topology. Segment Routing, including both SR MPLS and SRv6, are applications that MUST use such pruning when computing Flex-Algorithm paths.

When computing the path for a given Flex-Algorithm, the metric-type that is part of the Flex-Algorithm definition (Section 5) MUST be used.

When computing the path for a given Flex-Algorithm, the calculation-type that is part of the Flex-Algorithm definition (Section 5) MUST be used.

Various link include or exclude rules can be part of the Flex-Algorithm definition. To refer to a particular bit within an AG or EAG we use the term 'color'.

Rules, in the order as specified below, MUST be used to prune links from the topology during the Flex-Algorithm computation.

For all links in the topology:

1. Check if any exclude AG rule is part of the Flex-Algorithm definition. If such exclude rule exists, check if any color that is part of the exclude rule is also set on the link. If such a color is set, the link MUST be pruned from the computation.
2. Check if any exclude SRLG rule is part of the Flex-Algorithm definition. If such exclude rule exists, check if the link is part of any SRLG that is also part of the SRLG exclude rule. If the link is part of such SRLG, the link MUST be pruned from the computation.
3. Check if any include-any AG rule is part of the Flex-Algorithm definition. If such include-any rule exists, check if any color that is part of the include-any rule is also set on the link. If no such color is set, the link MUST be pruned from the computation.
4. Check if any include-all AG rule is part of the Flex-Algorithm definition. If such include-all rule exists, check if all colors that are part of the include-all rule are also set on the link. If all such colors are not set on the link, the link MUST be pruned from the computation.
5. If the Flex-Algorithm definition uses other than IGP metric (Section 5), and such metric is not advertised for the particular link in a topology for which the computation is done, such link

MUST be pruned from the computation. A metric of value 0 MUST NOT be assumed in such case.

13.1. Multi-area and Multi-domain Considerations

Any IGP Shortest Path Tree calculation is limited to a single area. This applies to Flex-Algorithm calculations as well. Given that the computing router does not have visibility of the topology of the next areas or domain, the Flex-Algorithm specific path to an inter-area or inter-domain prefix will be computed for the local area only. The egress L1/L2 router (ABR in OSPF), or ASBR for inter-domain case, will be selected based on the best path for the given Flex-Algorithm in the local area and such egress ABR or ASBR router will be responsible to compute the best Flex-Algorithm specific path over the next area or domain. This may produce an end-to-end path, which is sub-optimal based on Flex-Algorithm constraints. In cases where the ABR or ASBR has no reachability to a prefix for a given Flex-Algorithm in the next area or domain, the traffic may be dropped by the ABR/ASBR.

To allow the optimal end-to-end path for an inter-area or inter-domain prefix for any Flex-Algorithm to be computed, the FAPM has been defined in Section 8 and Section 9. For external route calculation for prefixes originated by ASBRs in remote areas in OSPF, the FAAM has been defined in Section 10.2 for the ABR to indicate its ASBR reachability along with the metric for the specific Flex-Algorithm.

If the FAD selected based on the rules defined in Section 5.3 includes the M-flag, an ABR or ASBR MUST include the FAPM (Section 8, Section 9) when advertising the prefix, that is reachable in a given Flex-Algorithm, between areas or domains. Such metric will be equal to the metric to reach the prefix for that Flex-Algorithm in its source area or domain. This is similar in nature to how the metric is set when prefixes are advertised between areas or domains for the default algorithm. When a prefix is unreachable in its source area or domain in a specific Flex-Algorithm, then an ABR or ASBR MUST NOT include the FAPM for that Flex-Algorithm when advertising the prefix between areas or domains.

If the FAD selected based on the rules defined in Section 5.3 includes the M-flag, the FAPM MUST be used during the calculation of prefix reachability for the inter-area and external prefixes. If the FAPM for the Flex-Algorithm is not advertised with the inter-area or external prefix reachability advertisement, the prefix MUST be considered as unreachable for that Flex-Algorithm. Similarly in the case of OSPF, for ASBRs in remote areas, if the FAAM is not advertised by the local ABR(s), the ASBR MUST be considered as

unreachable for that Flex-Algorithm and the external prefix advertisements from such an ASBR are not considered for that Flex-Algorithm.

Flex-Algorithm prefix metrics and the OSPF Flex-Algorithm ASBR metrics MUST NOT be used during the Flex-Algorithm computation unless the FAD selected based on the rules defined in Section 5.3 includes the M-Flag, as described in (Section 6.4 or Section 7.4).

In the case of OSPF, when calculating external routes in a Flex-Algorithm (with FAD selected includes the M-Flag) where the advertising ASBR is in a remote area, the metric will be the sum of the following:

- o the FAPM for that Flex-Algorithm advertised with the external route by the ASBR
- o the metric to reach the ASBR for that Flex-Algorithm from the local ABR i.e., the FAAM for that Flex-Algorithm advertised by the ABR in the local area for that ASBR
- o the Flex-Algorithm specific metric to reach the local ABR

This is similar in nature to how the metric is calculated for routes learned from remote ASBRs in the default algorithm using the OSPFv2 Type 4 ASBR Summary LSA and the OSPFv3 Inter-Area-Router LSA.

If the FAD selected based on the rules defined in Section 5.3 does not include the M-flag, then the IGP metrics associated with the prefix reachability advertisements used by the base IS-IS and OSPF protocol MUST be used for the Flex-Algorithm route computation. Similarly, in the case of external route calculations in OSPF, the ASBR reachability is determined based on the base OSPFv2 Type 4 Summary LSA and the OSPFv3 Inter-Area-Router LSA.

It is NOT RECOMMENDED to use the Flex-Algorithm for inter-area or inter-domain prefix reachability without the M-flag set. The reason is that without the explicit Flex-Algorithm Prefix Metric advertisement (and the Flex-Algorithm ASBR metric advertisement in the case of OSPF external route calculation), it is not possible to conclude whether the ABR or ASBR has reachability to the inter-area or inter-domain prefix for a given Flex-Algorithm in the next area or domain. Sending the Flex-Algorithm traffic for such prefix towards the ABR or ASBR may result in traffic looping or black-holing.

During the route computation, it is possible for the Flex-Algorithm specific metric to exceed the maximum value that can be stored in an unsigned 32-bit variable. In such scenarios, the value MUST be

considered to be of value 4,294,967,295 during the computation and advertised as such.

The FAPM MUST NOT be advertised with IS-IS L1 or L2 intra-area, OSPFv2 intra-area, or OSPFv3 intra-area routes. If the FAPM is advertised for these route-types, it MUST be ignored during the prefix reachability calculation.

The M-flag in FAD is not applicable to prefixes advertised as SRv6 locators. The IS-IS SRv6 Locator TLV [I-D.ietf-lsr-isis-srv6-extensions] includes the Algorithm and Metric fields. When the SRv6 Locator is advertised between areas or domains, the metric field in the Locator TLV of IS-IS MUST be used irrespective of the M-flag in the FAD advertisement.

OSPF external and NSSA external prefix advertisements MAY include a non-zero forwarding address in the prefix advertisements in the base protocol. In such a scenario, the Flex-Algorithm specific reachability of the external prefix is determined by Flex-Algorithm specific reachability of the forwarding address.

In OSPF, the procedures for translation of NSSA external prefix advertisements into external prefix advertisements performed by an NSSA ABR [RFC3101] remain unchanged for Flex-Algorithm. An NSSA translator MUST include the OSPF FAPM Sub-TLVs for all Flex-Algorithms that are in the original NSSA external prefix advertisement from the NSSA ASBR in the translated external prefix advertisement generated by it regardless of its participation in those Flex-Algorithms or its having reachability to the NSSA ASBR in those Flex-Algorithms.

An area could become partitioned from the perspective of the Flex-Algorithm due to the constraints and/or metric being used for it, while maintaining the continuity in the algorithm 0. When that happens, some destinations inside that area could become unreachable in that Flex-Algorithm. These destinations will not be able to use an inter-area path. This is the consequence of the fact that the inter-area prefix reachability advertisement would not be available for these intra-area destinations within the area. It is RECOMMENDED to avoid such partitioning by providing enough redundancy inside the area for each Flex-Algorithm being used.

14. Flex-Algorithm and Forwarding Plane

This section describes how Flex-Algorithm paths are used in forwarding.

14.1. Segment Routing MPLS Forwarding for Flex-Algorithm

This section describes how Flex-Algorithm paths are used with SR MPLS forwarding.

Prefix SID advertisements include an SR-Algorithm value and, as such, are associated with the specified SR-Algorithm. Prefix-SIDs are also associated with a specific topology which is inherited from the associated prefix reachability advertisement. When the algorithm value advertised is a Flex-Algorithm value, the Prefix SID is associated with paths calculated using that Flex-Algorithm in the associated topology.

A Flex-Algorithm path MUST be installed in the MPLS forwarding plane using the MPLS label that corresponds to the Prefix-SID that was advertised for that Flex-algorithm. If the Prefix SID for a given Flex-algorithm is not known, the Flex-Algorithm specific path cannot be installed in the MPLS forwarding plane.

Traffic that is supposed to be routed via Flex-Algorithm specific paths, MUST be dropped when there are no such paths available.

Loop Free Alternate (LFA) paths for a given Flex-Algorithm MUST be computed using the same constraints as the calculation of the primary paths for that Flex-Algorithm. LFA paths MUST only use Prefix-SIDs advertised specifically for the given algorithm. LFA paths MUST NOT use an Adjacency-SID that belongs to a link that has been pruned from the Flex-Algorithm computation.

If LFA protection is being used to protect a given Flex-Algorithm paths, all routers in the area participating in the given Flex-Algorithm SHOULD advertise at least one Flex-Algorithm specific Node-SID. These Node-SIDs are used to steer traffic over the LFA computed backup path.

14.2. SRv6 Forwarding for Flex-Algorithm

This section describes how Flex-Algorithm paths are used with SRv6 forwarding.

In SRv6 a node is provisioned with topology/algorithm specific locators for each of the topology/algorithm pairs supported by that node. Each locator is an aggregate prefix for all SIDs provisioned on that node which have the matching topology/algorithm.

The SRv6 locator advertisement in IS-IS [I-D.ietf-lsr-isis-srv6-extensions] includes the MTID value that associates the locator with a specific topology. SRv6 locator

advertisements also includes an Algorithm value that explicitly associates the locator with a specific algorithm. When the algorithm value advertised with a locator represents a Flex-Algorithm, the paths to the locator prefix MUST be calculated using the specified Flex-Algorithm in the associated topology.

Forwarding entries for the locator prefixes advertised in IS-IS MUST be installed in the forwarding plane of the receiving SRv6 capable routers when the associated topology/algorithm is participating in them. Forwarding entries for locators associated with Flex-Algorithms in which the node is not participating MUST NOT be installed in the forwarding plane.

When the locator is associated with a Flex-Algorithm, LFA paths to the locator prefix MUST be calculated using such Flex-Algorithm in the associated topology, to guarantee that they follow the same constraints as the calculation of the primary paths. LFA paths MUST only use SRv6 SIDs advertised specifically for the given Flex-Algorithm.

If LFA protection is being used to protect locators associated with a given Flex-Algorithm, all routers in the area participating in the given Flex-Algorithm SHOULD advertise at least one Flex-Algorithm specific locator and END SID per node and one END.X SID for every link that has not been pruned from such Flex-Algorithm computation. These locators and SIDs are used to steer traffic over the LFA-computed backup path.

14.3. Other Applications' Forwarding for Flex-Algorithm

Any application that wants to use Flex-Algorithm specific forwarding needs to install some form of Flex-Algorithm specific forwarding entries.

Application-specific forwarding for Flex-Algorithm MUST be defined for each application and is outside of the scope of this document.

15. Operational Considerations

15.1. Inter-area Considerations

The scope of the Flex-Algorithm computation is an area, so is the scope of the FAD. In IS-IS, the Router Capability TLV in which the FAD Sub-TLV is advertised MUST have the S-bit clear, which prevents it to be flooded outside of the level in which it was originated. Even though in OSPF the FAD Sub-TLV can be flooded in an RI LSA that has AS flooding scope, the FAD selection is performed for each individual area in which it is being used.

There is no requirement for the FAD for a particular Flex-Algorithm to be identical in all areas in the network. For example, traffic for the same Flex-Algorithm may be optimized for minimal delay (e.g., using delay metric) in one area or level, while being optimized for available bandwidth (e.g., using IGP metric) in another area or level.

As described in Section 5.1, IS-IS allows the re-generation of the winning FAD from level 2, without any modification to it, into a level 1 area. This allows the operator to configure the FAD in one or multiple routers in the level 2, without the need to repeat the same task in each level 1 area, if the intent is to have the same FAD for the particular Flex-Algorithm across all levels. This can similarly be achieved in OSPF by using the AS flooding scope of the RI LSA in which the FAD Sub-TLV for the particular Flex-Algorithm is advertised.

Re-generation of FAD from a level 1 area to the level 2 area is not supported in IS-IS, so if the intent is to regenerate the FAD between IS-IS levels, the FAD MUST be defined on router(s) that are in level 2. In OSPF, the FAD definition can be done in any area and be propagated to all routers in the OSPF routing domain by using the AS flooding scope of the RI LSA.

15.2. Usage of SRLG Exclude Rule with Flex-Algorithm

There are two different ways in which SRLG information can be used with Flex-Algorithm:

- In a context of a single Flex-Algorithm, it can be used for computation of backup paths, as described in [I-D.ietf-rtgwg-segment-routing-ti-lfa]. This usage does not require association of any specific SRLG constraint with the given Flex-Algorithm definition.

- In the context of multiple Flex-Algorithms, it can be used for creating disjoint sets of paths by pruning the links belonging to a specific SRLG from the topology on which a specific Flex-Algorithm computes its paths. This usage:

 - Facilitates the usage of already deployed SRLG configurations for setup of disjoint paths between two or more Flex-Algorithms.

 - Requires explicit association of a given Flex-Algorithm with a specific set of SRLG constraints as defined in Section 6.5 and Section 7.5.

The two usages mentioned above are orthogonal.

15.3. Max-metric consideration

Both IS-IS and OSPF have a mechanism to set the IGP metric on a link to a value that would make the link either non-reachable or to serve as the link of last resort. Similar functionality would be needed for the Min Unidirectional Link Delay and TE metric, as these can be used to compute Flex-Algorithm paths.

The link can be made un-reachable for all Flex-Algorithms that use Min Unidirectional Link Delay as metric, as described in Section 5.1, by removing the Flex-Algorithm ASLA Min Unidirectional Link Delay advertisement for the link. The link can be made the link of last resort by setting the delay value in the Flex-Algorithm ASLA delay advertisement for the link to the value of 16,777,215 ($2^{24} - 1$).

The link can be made un-reachable for all Flex-Algorithms that use TE metric, as described in Section 5.1, by removing the Flex-Algorithm ASLA TE metric advertisement for the link. The link can be made the link of last resort by setting the TE metric value in the Flex-Algorithm ASLA delay advertisement for the link to the value of ($2^{24} - 1$) in IS-IS and ($2^{32} - 1$) in OSPF.

16. Backward Compatibility

This extension brings no new backward compatibility issues. IS-IS, OSPFv2 and OSPFv3 all have well defined handling of unrecognized TLVs and sub-TLVs that allows the introduction of the new extensions, similar to those defined here, without introducing any interoperability issues.

17. Security Considerations

This draft adds two new ways to disrupt IGP networks:

An attacker can hijack a particular Flex-Algorithm by advertising a FAD with a priority of 255 (or any priority higher than that of the legitimate nodes).

An attacker could make it look like a router supports a particular Flex-Algorithm when it actually doesn't, or vice versa.

Both of these attacks can be addressed by the existing security extensions as described in [RFC5304] and [RFC5310] for IS-IS, in [RFC2328] and [RFC7474] for OSPFv2, and in [RFC5340] and [RFC4552] for OSPFv3.

18. IANA Considerations

18.1. IGP IANA Considerations

18.1.1. IGP Algorithm Types Registry

This document makes the following registrations in the "IGP Algorithm Types" registry:

Type: 128-255.

Description: Flexible Algorithms.

Reference: This document (Section 4).

18.1.2. IGP Metric-Type Registry

IANA is requested to set up a registry called "IGP Metric-Type Registry" under an "Interior Gateway Protocol (IGP) Parameters" IANA registries. The registration policy for this registry is "Standards Action" ([RFC8126] and [RFC7120]).

Values in this registry come from the range 0-255.

This document registers following values in the "IGP Metric-Type Registry":

Type: 0

Description: IGP metric

Reference: This document (Section 5.1)

Type: 1

Description: Min Unidirectional Link Delay as defined in [RFC8570], section 4.2, and [RFC7471], section 4.2.

Reference: This document (Section 5.1)

Type: 2

Description: Traffic Engineering Default Metric as defined in [RFC5305], section 3.7, and Traffic engineering metric as defined in [RFC3630], section 2.5.5

Reference: This document (Section 5.1)

18.2. Flexible Algorithm Definition Flags Registry

IANA is requested to set up a registry called "IS-IS Flexible Algorithm Definition Flags Registry" under an "Interior Gateway Protocol (IGP) Parameters" IANA registries. The registration policy for this registry is "Standards Action" ([RFC8126] and [RFC7120]).

This document defines the following single bit in Flexible Algorithm Definition Flags registry:

Bit #	Name
-----	-----
0	Prefix Metric Flag (M-flag)

Reference: This document (Section 6.4, Section 7.4).

18.3. IS-IS IANA Considerations

18.3.1. Sub TLVs for Type 242

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

Type: 26.

Description: Flexible Algorithm Definition.

Reference: This document (Section 5.1).

18.3.2. Sub TLVs for for TLVs 135, 235, 236, and 237

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

Type: 6

Description: Flexible Algorithm Prefix Metric.

Reference: This document (Section 8).

18.3.3. Sub-Sub-TLVs for Flexible Algorithm Definition Sub-TLV

This document creates the following Sub-Sub-TLV Registry:

Registry: Sub-Sub-TLVs for Flexible Algorithm Definition Sub-TLV

Registration Procedure: Expert review

Reference: This document (Section 5.1)

This document defines the following Sub-Sub-TLVs in the "Sub-Sub-TLVs for Flexible Algorithm Definition Sub-TLV" registry:

Type: 1

Description: Flexible Algorithm Exclude Admin Group

Reference: This document (Section 6.1).

Type: 2

Description: Flexible Algorithm Include-Any Admin Group

Reference: This document (Section 6.2).

Type: 3

Description: Flexible Algorithm Include-All Admin Group

Reference: This document (Section 6.3).

Type: 4

Description: Flexible Algorithm Definition Flags

Reference: This document (Section 6.4).

Type: 5

Description: Flexible Algorithm Exclude SRLG

Reference: This document (Section 6.5).

18.4. OSPF IANA Considerations

18.4.1. OSPF Router Information (RI) TLVs Registry

This specification updates the OSPF Router Information (RI) TLVs Registry.

Type: 16

Description: Flexible Algorithm Definition TLV.

Reference: This document (Section 5.2).

18.4.2. OSPFv2 Extended Prefix TLV Sub-TLVs

This document makes the following registrations in the "OSPFv2 Extended Prefix TLV Sub-TLVs" registry.

Type: 3

Description: Flexible Algorithm Prefix Metric.

Reference: This document (Section 9).

18.4.3. OSPFv3 Extended-LSA Sub-TLVs

This document makes the following registrations in the "OSPFv3 Extended-LSA Sub-TLVs" registry.

Type: 26

Description: Flexible Algorithm Prefix Metric.

Reference: This document (Section 9).

Type: TBD (suggested value 30)

Description: OSPF Flexible Algorithm ASBR Metric Sub-TLV

Reference: This document (Section 10.2).

18.4.4. OSPF Flex-Algorithm Prefix Metric Bits

This specification requests creation of "OSPF Flex-Algorithm Prefix Metric Bits" registry under the OSPF Parameters Registry with the following initial values.

Bit Number: 0

Description: E bit - External Type

Reference: this document.

The bits 1-7 are unassigned and the registration procedure to be followed for this registry is IETF Review.

18.4.5. OSPF Opaque LSA Option Types

This document makes the following registrations in the "OSPF Opaque LSA Option Types" registry.

Value: TBD (suggested value 11)

Description: OSPFv2 Extended Inter-Area ASBR LSA

Reference: This document (Section 10.1).

18.4.6. OSPFv2 Extended Inter-Area ASBR TLVs

This specification requests creation of "OSPFv2 Extended Inter-Area ASBR TLVs" registry under the OSPFv2 Parameters Registry with the following initial values.

Value: 1

Description : Extended Inter-Area ASBR TLV

Reference: this document

The values 2 to 32767 are unassigned, values 32768 to 33023 are reserved for experimental use while the values 0 and 33024 to 65535 are reserved. The registration procedure to be followed for this registry is IETF Review or IESG Approval.

18.4.7. OSPFv2 Inter-Area ASBR Sub-TLVs

This specification requests creation of "OSPFv2 Extended Inter-Area ASBR Sub-TLVs" registry under the OSPFv2 Parameters Registry with the following initial values.

Value: 1

Description : OSPF Flexible Algorithm ASBR Metric Sub-TLV

Reference: this document

The values 2 to 32767 are unassigned, values 32768 to 33023 are reserved for experimental use while the values 0 and 33024 to 65535 are reserved. The registration procedure to be followed for this registry is IETF Review or IESG Approval.

18.4.8. OSPF Flexible Algorithm Definition TLV Sub-TLV Registry

This document creates the following registry:

Registry: OSPF Flexible Algorithm Definition TLV sub-TLV

Registration Procedure: Expert review

Reference: This document (Section 5.2)

The "OSPF Flexible Algorithm Definition TLV sub-TLV" registry will define sub-TLVs at any level of nesting for the Flexible Algorithm TLV and should be added to the "Open Shortest Path First (OSPF) Parameters" registries group. New values can be allocated via IETF Review or IESG Approval.

This document registers following Sub-TLVs in the "TLVs for Flexible Algorithm Definition TLV" registry:

Type: 1

Description: Flexible Algorithm Exclude Admin Group

Reference: This document (Section 7.1).

Type: 2

Description: Flexible Algorithm Include-Any Admin Group

Reference: This document (Section 7.2).

Type: 3

Description: Flexible Algorithm Include-All Admin Group

Reference: This document (Section 7.3).

Type: 4

Description: Flexible Algorithm Definition Flags

Reference: This document (Section 7.4).

Type: 5

Description: Flexible Algorithm Exclude SRLG

Reference: This document (Section 7.5).

Types in the range 32768-33023 are for experimental use; these will not be registered with IANA, and MUST NOT be mentioned by RFCs.

Types in the range 33024-65535 are not to be assigned at this time. Before any assignments can be made in the 33024-65535 range, there MUST be an IETF specification that specifies IANA Considerations that covers the range being assigned.

18.4.9. Link Attribute Applications Registry

This document registers following bit in the Link Attribute Applications Registry:

Bit-3

Description: Flexible Algorithm (X-bit)

Reference: This document (Section 12).

19. Acknowledgements

This draft, among other things, is also addressing the problem that the [I-D.gulkohegde-routing-planes-using-sr] was trying to solve. All authors of that draft agreed to join this draft.

Thanks to Eric Rosen, Tony Przygienda, William Britto A J, Gunter Van De Velde, Dirk Goethals, Manju Sivaji and, Baalajee S for their detailed review and excellent comments.

Thanks to Cengiz Halit for his review and feedback during initial phase of the solution definition.

Thanks to Kenji Kumaki for his comments.

Thanks to Acee Lindem for editorial comments.

20. References

20.1. Normative References

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

Psenak, P., Filsfils, C., Bashandy, A., Decraene, B., and Z. Hu, "IS-IS Extensions to Support Segment Routing over IPv6 Dataplane", draft-ietf-lsr-isis-srv6-extensions-18 (work in progress), October 2021.

[ISO10589]

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

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC4203] Kompella, K., Ed. and Y. Rekhter, Ed., "OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4203, DOI 10.17487/RFC4203, October 2005, <<https://www.rfc-editor.org/info/rfc4203>>.
- [RFC5250] Berger, L., Bryskin, I., Zinin, A., and R. Coltun, "The OSPF Opaque LSA Option", RFC 5250, DOI 10.17487/RFC5250, July 2008, <<https://www.rfc-editor.org/info/rfc5250>>.
- [RFC5307] Kompella, K., Ed. and Y. Rekhter, Ed., "IS-IS Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 5307, DOI 10.17487/RFC5307, October 2008, <<https://www.rfc-editor.org/info/rfc5307>>.
- [RFC7308] Osborne, E., "Extended Administrative Groups in MPLS Traffic Engineering (MPLS-TE)", RFC 7308, DOI 10.17487/RFC7308, July 2014, <<https://www.rfc-editor.org/info/rfc7308>>.
- [RFC7684] Psenak, P., Gredler, H., Shakir, R., Henderickx, W., Tantsura, J., and A. Lindem, "OSPFv2 Prefix/Link Attribute Advertisement", RFC 7684, DOI 10.17487/RFC7684, November 2015, <<https://www.rfc-editor.org/info/rfc7684>>.
- [RFC7770] Lindem, A., Ed., Shen, N., Vasseur, JP., Aggarwal, R., and S. Shaffer, "Extensions to OSPF for Advertising Optional Router Capabilities", RFC 7770, DOI 10.17487/RFC7770, February 2016, <<https://www.rfc-editor.org/info/rfc7770>>.
- [RFC7981] Ginsberg, L., Previdi, S., and M. Chen, "IS-IS Extensions for Advertising Router Information", RFC 7981, DOI 10.17487/RFC7981, October 2016, <<https://www.rfc-editor.org/info/rfc7981>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8362] Lindem, A., Roy, A., Goethals, D., Reddy Vallem, V., and F. Baker, "OSPFv3 Link State Advertisement (LSA) Extensibility", RFC 8362, DOI 10.17487/RFC8362, April 2018, <<https://www.rfc-editor.org/info/rfc8362>>.

- [RFC8665] Psenak, P., Ed., Previdi, S., Ed., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", RFC 8665, DOI 10.17487/RFC8665, December 2019, <<https://www.rfc-editor.org/info/rfc8665>>.
- [RFC8666] Psenak, P., Ed. and S. Previdi, Ed., "OSPFv3 Extensions for Segment Routing", RFC 8666, DOI 10.17487/RFC8666, December 2019, <<https://www.rfc-editor.org/info/rfc8666>>.
- [RFC8667] Previdi, S., Ed., Ginsberg, L., Ed., Filsfils, C., Bashandy, A., Gredler, H., and B. Decraene, "IS-IS Extensions for Segment Routing", RFC 8667, DOI 10.17487/RFC8667, December 2019, <<https://www.rfc-editor.org/info/rfc8667>>.
- [RFC8919] Ginsberg, L., Psenak, P., Previdi, S., Henderickx, W., and J. Drake, "IS-IS Application-Specific Link Attributes", RFC 8919, DOI 10.17487/RFC8919, October 2020, <<https://www.rfc-editor.org/info/rfc8919>>.
- [RFC8920] Psenak, P., Ed., Ginsberg, L., Henderickx, W., Tantsura, J., and J. Drake, "OSPF Application-Specific Link Attributes", RFC 8920, DOI 10.17487/RFC8920, October 2020, <<https://www.rfc-editor.org/info/rfc8920>>.

20.2. Informative References

- [I-D.gulkohegde-routing-planes-using-sr] Hegde, S. and A. Gulko, "Separating Routing Planes using Segment Routing", draft-gulkohegde-routing-planes-using-sr-00 (work in progress), March 2017.
- [I-D.ietf-rtgwg-segment-routing-ti-lfa] Litkowski, S., Bashandy, A., Filsfils, C., Francois, P., Decraene, B., and D. Voyer, "Topology Independent Fast Reroute using Segment Routing", draft-ietf-rtgwg-segment-routing-ti-lfa-08 (work in progress), January 2022.
- [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <<https://www.rfc-editor.org/info/rfc2328>>.
- [RFC3101] Murphy, P., "The OSPF Not-So-Stubby Area (NSSA) Option", RFC 3101, DOI 10.17487/RFC3101, January 2003, <<https://www.rfc-editor.org/info/rfc3101>>.

- [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, DOI 10.17487/RFC3630, September 2003, <<https://www.rfc-editor.org/info/rfc3630>>.
- [RFC3906] Shen, N. and H. Smit, "Calculating Interior Gateway Protocol (IGP) Routes Over Traffic Engineering Tunnels", RFC 3906, DOI 10.17487/RFC3906, October 2004, <<https://www.rfc-editor.org/info/rfc3906>>.
- [RFC4552] Gupta, M. and N. Melam, "Authentication/Confidentiality for OSPFv3", RFC 4552, DOI 10.17487/RFC4552, June 2006, <<https://www.rfc-editor.org/info/rfc4552>>.
- [RFC5304] Li, T. and R. Atkinson, "IS-IS Cryptographic Authentication", RFC 5304, DOI 10.17487/RFC5304, October 2008, <<https://www.rfc-editor.org/info/rfc5304>>.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <<https://www.rfc-editor.org/info/rfc5305>>.
- [RFC5310] Bhatia, M., Manral, V., Li, T., Atkinson, R., White, R., and M. Fanto, "IS-IS Generic Cryptographic Authentication", RFC 5310, DOI 10.17487/RFC5310, February 2009, <<https://www.rfc-editor.org/info/rfc5310>>.
- [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <<https://www.rfc-editor.org/info/rfc5340>>.
- [RFC7120] Cotton, M., "Early IANA Allocation of Standards Track Code Points", BCP 100, RFC 7120, DOI 10.17487/RFC7120, January 2014, <<https://www.rfc-editor.org/info/rfc7120>>.
- [RFC7471] Giacalone, S., Ward, D., Drake, J., Atlas, A., and S. Previdi, "OSPF Traffic Engineering (TE) Metric Extensions", RFC 7471, DOI 10.17487/RFC7471, March 2015, <<https://www.rfc-editor.org/info/rfc7471>>.
- [RFC7474] Bhatia, M., Hartman, S., Zhang, D., and A. Lindem, Ed., "Security Extension for OSPFv2 When Using Manual Key Management", RFC 7474, DOI 10.17487/RFC7474, April 2015, <<https://www.rfc-editor.org/info/rfc7474>>.

- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [RFC8570] Ginsberg, L., Ed., Previdi, S., Ed., Giacalone, S., Ward, D., Drake, J., and Q. Wu, "IS-IS Traffic Engineering (TE) Metric Extensions", RFC 8570, DOI 10.17487/RFC8570, March 2019, <<https://www.rfc-editor.org/info/rfc8570>>.

Authors' Addresses

Peter Psenak (editor)
Cisco Systems
Apollo Business Center
Mlynske nivy 43
Bratislava, 82109
Slovakia

Email: ppsenak@cisco.com

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

Email: shraddha@juniper.net

Clarence Filsfils
Cisco Systems, Inc.
Brussels
Belgium

Email: cfilsfil@cisco.com

Ketan Talaulikar
Arrcus, Inc
India

Email: ketant.ietf@gmail.com

Arkadiy Gulko
Edward Jones

Email: arkadiy.gulko@edwardjones.com

LSR Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 3, 2020

X. Xu
Alibaba Inc
S. Kini

P. Psenak
C. Filsfils
S. Litkowski
Cisco Systems, Inc.
M. Bocci
Nokia
June 1, 2020

Signaling Entropy Label Capability and Entropy Readable Label Depth
Using OSPF
draft-ietf-ospf-mpls-elc-15

Abstract

Multiprotocol Label Switching (MPLS) has defined a mechanism to load-balance traffic flows using Entropy Labels (EL). An ingress Label Switching Router (LSR) cannot insert ELs for packets going into a given Label Switched Path (LSP) unless an egress LSR has indicated via signaling that it has the capability to process ELs, referred to as the Entropy Label Capability (ELC), on that LSP. In addition, it would be useful for ingress LSRs to know each LSR's capability for reading the maximum label stack depth and performing EL-based load-balancing, referred to as Entropy Readable Label Depth (ERLD). This document defines a mechanism to signal these two capabilities using OSPFv2 and OSPFv3 and BGP-LS.

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 3, 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. Advertising ELC Using OSPF	3
3.1. Advertising ELC Using OSPFv2	3
3.2. Advertising ELC Using OSPFv3	4
4. Advertising ERLD Using OSPF	4
5. Signaling ELC and ERLD in BGP-LS	5
6. IANA Considerations	5
7. Security Considerations	5
8. Contributors	6
9. Acknowledgements	6
10. References	6
10.1. Normative References	6
10.2. Informative References	8
Authors' Addresses	8

1. Introduction

[RFC6790] describes a method to load-balance Multiprotocol Label Switching (MPLS) traffic flows using Entropy Labels (EL). It also introduces the concept of Entropy Label Capability (ELC) and defines the signaling of this capability via MPLS signaling protocols. Recently, mechanisms have been defined to signal labels via link-state Interior Gateway Protocols (IGP) such as OSPFv2 [RFC8665] and OSPFv3 [RFC8666]. This draft defines a mechanism to signal the ELC using OSPFv2 and OSPFv3.

In cases where Segment Routing (SR) is used with the MPLS Data Plane (e.g., SR-MPLS [RFC8660]), it would be useful for ingress LSRs to know each intermediate LSR's capability of reading the maximum label stack depth and performing EL-based load-balancing. This capability,

referred to as Entropy Readable Label Depth (ERLD) as defined in [RFC8662], may be used by ingress LSRs to determine the position of the EL label in the stack, and whether it is necessary to insert multiple ELs at different positions in the label stack. This document defines a mechanism to signal the ERLD using OSPFv2 and OSPFv3.

2. Terminology

This memo makes use of the terms defined in [RFC6790], and [RFC8662].

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.

The key word OSPF is used throughout the document to refer to both OSPFv2 and OSPFv3.

3. Advertising ELC Using OSPF

Even though ELC is a property of the node, in some cases it is advantageous to associate and advertise the ELC with a prefix. In multi-area networks, routers may not know the identity of the prefix originator in a remote area, or may not know the capabilities of such originator. Similarly, in a multi domain network, the identity of the prefix originator and its capabilities may not be known to the ingress LSR.

If a router has multiple interfaces, the router MUST NOT announce ELC unless all of its interfaces are capable of processing ELs.

If the router supports ELs on all of its interfaces, it SHOULD advertise the ELC with every local host prefix it advertises in OSPF.

3.1. Advertising ELC Using OSPFv2

[RFC7684] defines the OSPFv2 Extended Prefix TLV to advertise additional attributes associated with a prefix. The OSPFv2 Extended Prefix TLV includes a one-octet Flags field. A new flag in the Flags field is used to signal the ELC for the prefix:

0x20 - E-Flag (ELC Flag): Set by the advertising router to indicate that the prefix originator is capable of processing ELs.

The ELC signaling MUST be preserved when an OSPF Area Border Router (ABR) distributes information between areas. To do so, an ABR MUST

originate an OSPFv2 Extended Prefix Opaque LSA [RFC7684] including the received ELC setting.

When an OSPF Autonomous System Boundary Router (ASBR) redistributes a prefix from another instance of OSPF or from some other protocol, it SHOULD preserve the ELC signaling for the prefix if it exists. To do so, an ASBR SHOULD originate an Extended Prefix Opaque LSA [RFC7684] including the ELC setting of the redistributed prefix. The flooding scope of the Extended Prefix Opaque LSA MUST match the flooding scope of the LSA that an ASBR originates as a result of the redistribution. The exact mechanism used to exchange ELC between protocol instances on an ASBR is outside of the scope of this document.

3.2. Advertising ELC Using OSPFv3

[RFC5340] defines the OSPFv3 PrefixOptions field to indicate capabilities associated with a prefix. A new bit in the OSPFv3 PrefixOptions is used to signal the ELC for the prefix:

0x40 - E-Flag (ELC Flag): Set by the advertising router to indicate that the prefix originator is capable of processing ELs.

The ELC signaling MUST be preserved when an OSPFv3 Area Border Router (ABR) distributes information between areas. The setting of the ELC Flag in the Inter-Area-Prefix-LSA [RFC5340] or in the Inter-Area-Prefix TLV [RFC8362], generated by an ABR, MUST be the same as the value the ELC Flag associated with the prefix in the source area.

When an OSPFv3 Autonomous System Boundary Router (ASBR) redistributes a prefix from another instance of OSPFv3 or from some other protocol, it SHOULD preserve the ELC signaling for the prefix if it exists. The setting of the ELC Flag in the AS-External-LSA, NSSA-LSA [RFC5340] or in the External-Prefix TLV [RFC8362], generated by an ASBR, MUST be the same as the value of the ELC Flag associated with the prefix in the source domain. The exact mechanism used to exchange ELC between protocol instances on the ASBR is outside of the scope of this document.

4. Advertising ERLD Using OSPF

The ERLD is advertised in a Node MSD TLV [RFC8476] using the ERLD-MSD type defined in [I-D.ietf-isis-mpls-eld].

If a router has multiple interfaces with different capabilities of reading the maximum label stack depth, the router MUST advertise the smallest value found across all of its interfaces.

The absence of ERLD-MSD advertisements indicates only that the advertising node does not support advertisement of this capability.

When the ERLD-MSD type is received in the OSPFv2 or OSPFv3 Link MSD Sub-TLV [RFC8476], it MUST be ignored.

The considerations for advertising the ERLD are specified in [RFC8662].

5. Signaling ELC and ERLD in BGP-LS

The OSPF extensions defined in this document can be advertised via BGP-LS (Distribution of Link-State and TE Information Using BGP) [RFC7752] using existing BGP-LS TLVs.

The ELC is advertised using the Prefix Attribute Flags TLV as defined in [I-D.ietf-idr-bgp-ls-segment-routing-ext].

The ERLD-MSD is advertised using the Node MSD TLV as defined in [I-D.ietf-idr-bgp-ls-segment-routing-msd].

6. IANA Considerations

Early allocation has been done by IANA for this document as follows:

- Flag 0x20 in the OSPFv2 Extended Prefix TLV Flags registry has been allocated by IANA to the E-Flag (ELC Flag).
- Bit 0x40 in the "OSPFv3 Prefix Options (8 bits)" registry has been allocated by IANA to the E-Flag (ELC Flag).

7. Security Considerations

This document specifies the ability to advertise additional node capabilities using OSPF and BGP-LS. As such, the security considerations as described in [RFC5340], [RFC7770], [RFC7752], [RFC7684], [RFC8476], [RFC8662], [I-D.ietf-idr-bgp-ls-segment-routing-ext] and [I-D.ietf-idr-bgp-ls-segment-routing-msd] are applicable to this document.

Incorrectly setting the E flag during origination, propagation or redistribution may lead to poor or no load-balancing of the MPLS traffic or black-holing of the MPLS traffic on the egress node.

Incorrectly setting of the ERLD value may lead to poor or no load-balancing of the MPLS traffic.

8. Contributors

The following people contributed to the content of this document and should be considered as co-authors:

Gunter Van de Velde (editor)
Nokia
Antwerp
BE

Email: gunter.van_de_velde@nokia.com

Wim Henderickx
Nokia
Belgium

Email: wim.henderickx@nokia.com

Keyur Patel
Arrcus
USA

Email: keyur@arrcus.com

9. Acknowledgements

The authors would like to thank Yimin Shen, George Swallow, Acee Lindem, Les Ginsberg, Ketan Talaulikar, Jeff Tantsura, Bruno Decraene and Carlos Pignataro for their valuable comments.

10. References

10.1. Normative References

[I-D.ietf-idr-bgp-ls-segment-routing-ext]
Previdi, S., Talaulikar, K., Filsfils, C., Gredler, H.,
and M. Chen, "BGP Link-State extensions for Segment
Routing", draft-ietf-idr-bgp-ls-segment-routing-ext-16
(work in progress), June 2019.

- [I-D.ietf-idr-bgp-ls-segment-routing-msd]
Tantsura, J., Chunduri, U., Talaulikar, K., Mirsky, G.,
and N. Triantafyllis, "Signaling MSD (Maximum SID Depth)
using Border Gateway Protocol - Link State", draft-ietf-
idr-bgp-ls-segment-routing-msd-18 (work in progress), May
2020.
- [I-D.ietf-isis-mpls-elc]
Xu, X., Kini, S., Psenak, P., Filsfils, C., Litkowski, S.,
and M. Bocci, "Signaling Entropy Label Capability and
Entropy Readable Label Depth Using IS-IS", draft-ietf-
isis-mpls-elc-13 (work in progress), May 2020.
- [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>>.
- [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>>.
- [RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and
L. Yong, "The Use of Entropy Labels in MPLS Forwarding",
RFC 6790, DOI 10.17487/RFC6790, November 2012,
<<https://www.rfc-editor.org/info/rfc6790>>.
- [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>>.
- [RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and
S. Ray, "North-Bound Distribution of Link-State and
Traffic Engineering (TE) Information Using BGP", RFC 7752,
DOI 10.17487/RFC7752, March 2016,
<<https://www.rfc-editor.org/info/rfc7752>>.
- [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>>.
- [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>>.
- [RFC8476] Tantsura, J., Chunduri, U., Aldrin, S., and P. Psenak, "Signaling Maximum SID Depth (MSD) Using OSPF", RFC 8476, DOI 10.17487/RFC8476, December 2018, <<https://www.rfc-editor.org/info/rfc8476>>.
- [RFC8662] Kini, S., Kompella, K., Sivabalan, S., Litkowski, S., Shakir, R., and J. Tantsura, "Entropy Label for Source Packet Routing in Networking (SPRING) Tunnels", RFC 8662, DOI 10.17487/RFC8662, December 2019, <<https://www.rfc-editor.org/info/rfc8662>>.

10.2. Informative References

- [RFC8660] Bashandy, A., Ed., Filsfils, C., Ed., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with the MPLS Data Plane", RFC 8660, DOI 10.17487/RFC8660, December 2019, <<https://www.rfc-editor.org/info/rfc8660>>.
- [RFC8665] Psenak, P., Ed., Previdi, S., Ed., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", RFC 8665, DOI 10.17487/RFC8665, December 2019, <<https://www.rfc-editor.org/info/rfc8665>>.
- [RFC8666] Psenak, P., Ed. and S. Previdi, Ed., "OSPFv3 Extensions for Segment Routing", RFC 8666, DOI 10.17487/RFC8666, December 2019, <<https://www.rfc-editor.org/info/rfc8666>>.

Authors' Addresses

Xiaohu Xu
Alibaba Inc

Email: xiaohu.xxh@alibaba-inc.com

Sriganesh Kini

Email: sriganeshkini@gmail.com

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

Email: ppsenak@cisco.com

Clarence Filsfils
Cisco Systems, Inc.
Brussels
Belgium

Email: cfilsfil@cisco.com

Stephane Litkowski
Cisco Systems, Inc.
La Rigourdiere
Cesson Sevigne
France

Email: slitkows@cisco.com

Matthew Bocci
Nokia
Shoppenhangers Road
Maidenhead, Berks
UK

Email: matthew.bocci@nokia.com

Internet
Internet-Draft
Intended status: Standards Track
Expires: 6 July 2022

D. Yeung
Arrcus
Y. Qu
Futurewei
J. Zhang
Juniper Networks
I. Chen
The MITRE Corporation
A. Lindem
Cisco Systems
2 January 2022

YANG Data Model for OSPF Segment Routing
draft-ietf-ospf-sr-yang-17

Abstract

This document defines a YANG data module that can be used to configure and manage OSPF Extensions for Segment Routing. It also defines a module for management of Signaling Maximum SID Depth (MSD) Using OSPF.

Status of This Memo

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

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

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

This Internet-Draft will expire on 6 July 2022.

Copyright Notice

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

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

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

Table of Contents

1. Overview	2
1.1. Requirements Language	3
1.2. Tree Diagrams	3
2. OSPF MSD	3
2.1. OSPF MSD YANG Module	4
3. OSPF Segment Routing	11
3.1. OSPF Segment Routing YANG Module	16
4. Security Considerations	30
5. Acknowledgements	31
6. IANA Considerations	31
7. References	32
7.1. Normative References	32
7.2. Informative References	34
Appendix A. Contributors' Addresses	34
Authors' Addresses	34

1. Overview

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

This document defines a YANG data model that can be used to configure and manage OSPFv2 extensions for Segment Routing [RFC8665] and it is an augmentation to the OSPF YANG data model.

This document also defines a YANG data model for Signaling Maximum SID Depth (MSD) Using OSPF [RFC8476], which augments the base OSPF YANG data model.

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

1.1. Requirements Language

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

1.2. Tree Diagrams

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

2. OSPF MSD

This document defines a model for Signaling Maximum SID Depth (MSD) Using OSPF [RFC8476]. It is an augmentation of the OSPF base model.

```

module: ietf-ospf-msd
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
    /ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
    /ospf:area-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
    /ospf:body/ospf:opaque/ospf:ri-opaque:
  +--ro node-msd-tlv
    +--ro node-msds* [msd-type]
      +--ro msd-type      identityref
      +--ro msd-value?    uint8
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/ospf:ospf/ospf:database
    /ospf:as-scope-lsa-type/ospf:as-scope-lsas/ospf:as-scope-lsa
    /ospf:version/ospf:ospfv2/ospf:ospfv2/ospf:body/ospf:opaque
    /ospf:ri-opaque:
  +--ro node-msd-tlv
    +--ro node-msds* [msd-type]
      +--ro msd-type      identityref
      +--ro msd-value?    uint8
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
    /ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
    /ospf:area-scope-lsa/ospf:version/ospf:ospfv3/ospf:ospfv3
    /ospf:body/ospf:router-information:
  +--ro node-msd-tlv
    +--ro node-msds* [msd-type]
      +--ro msd-type      identityref
      +--ro msd-value?    uint8
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/ospf:ospf/ospf:database
    /ospf:as-scope-lsa-type/ospf:as-scope-lsas/ospf:as-scope-lsa
    /ospf:version/ospf:ospfv3/ospf:ospfv3/ospf:body

```

```

        /ospf:router-information:
+--ro node-msd-tlv
  +--ro node-msds* [msd-type]
    +--ro msd-type      identityref
    +--ro msd-value?    uint8
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
  /ospf:interfaces/ospf:interface/ospf:database
  /ospf:link-scope-lsa-type/ospf:link-scope-lsas
  /ospf:link-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
  /ospf:body/ospf:opaque/ospf:extended-link-opaque
  /ospf:extended-link-tlv:
+--ro link-msd-sub-tlv
  +--ro link-msds* [msd-type]
    +--ro msd-type      identityref
    +--ro msd-value?    uint8
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
  /ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
  /ospf:area-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
  /ospf:body/ospf:opaque/ospf:extended-link-opaque
  /ospf:extended-link-tlv:
+--ro link-msd-sub-tlv
  +--ro link-msds* [msd-type]
    +--ro msd-type      identityref
    +--ro msd-value?    uint8
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/ospf:ospf/ospf:database
  /ospf:as-scope-lsa-type/ospf:as-scope-lsas/ospf:as-scope-lsa
  /ospf:version/ospf:ospfv2/ospf:ospfv2/ospf:body/ospf:opaque
  /ospf:extended-link-opaque/ospf:extended-link-tlv:
+--ro link-msd-sub-tlv
  +--ro link-msds* [msd-type]
    +--ro msd-type      identityref
    +--ro msd-value?    uint8
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
  /ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
  /ospf:area-scope-lsa/ospf:version/ospf:ospfv3/ospf:ospfv3
  /ospf:body/ospfv3-e-lsa:e-router/ospfv3-e-lsa:e-router-tlvs:
+--ro link-msd-sub-tlv
  +--ro link-msds* [msd-type]
    +--ro msd-type      identityref
    +--ro msd-value?    uint8

```

2.1. OSPF MSD YANG Module

```
<CODE BEGINS> file "ietf-ospf-msd@2022-01-02.yang"
module ietf-ospf-msd {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-ospf-msd";
  prefix ospf-msd;

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

  import ietf-ospf {
    prefix ospf;
  }

  import ietf-ospfv3-extended-lsa {
    prefix ospfv3-e-lsa;
  }

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

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

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

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

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

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

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

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

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

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

revision 2022-01-02 {
  description
    "Initial Version";
  reference "RFC XXXX: YANG Data Model for OSPF MSD.";
}

identity msd-base-type {
  description
    "Base identity for MSD Type";
}

identity base-mpls-msd {
  base msd-base-type;
  description
    "Base MPLS Imposition MSD.";
  reference
    "RFC 8491: Singling MSD using IS-IS.";
}

identity erld-msd {
  base msd-base-type;
  description
    "ERLD-MSD is defined to advertise the ERLD.";
  reference
    "RFC 8662: Entropy Label for Source Packet Routing in
      Networking (SPRING) Tunnels";
}

grouping node-msd-tlv {
  description
    "Grouping for node MSD.";
```

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

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



```

        "Link MSD sub-tlvs.";
    }
}

/* Node MSD TLV */
augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:areas/"
+ "ospf:area/ospf:database/"
+ "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
+ "ospf:area-scope-lsa/ospf:version/ospf:ospfv2/"
+ "ospf:ospfv2/ospf:body/ospf:opaque/"
+ "ospf:ri-opaque" {
when "../../../../../../../../../../../../../../../"
+ "rt:type = 'ospf:ospfv2'" {
description
    "This augmentation is only valid for OSPFv2.";
}
description
    "Node MSD TLV is an optional TLV of OSPFv2 RI Opaque
    LSA (RFC7770) and has a type of 12.";

uses node-msd-tlv;
}

augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:database/"
+ "ospf:as-scope-lsa-type/ospf:as-scope-lsas/"
+ "ospf:as-scope-lsa/ospf:version/ospf:ospfv2/"
+ "ospf:ospfv2/ospf:body/ospf:opaque/"
+ "ospf:ri-opaque" {
when "../../../../../../../../../../../../../../../"
+ "rt:type = 'ospf:ospfv2'" {
description
    "This augmentation is only valid for OSPFv2.";
}
description
    "Node MSD TLV is an optional TLV of OSPFv2 RI Opaque
    LSA (RFC7770) and has a type of 12.";

uses node-msd-tlv;
}

augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:areas/"
+ "ospf:area/ospf:database/"

```

```

        + "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
        + "ospf:area-scope-lsa/ospf:version/ospf:ospfv3/"
        + "ospf:ospfv3/ospf:body/ospf:router-information" {
when "../.../.../.../.../.../.../.../.../..."
    + "rt:type = 'ospf:ospfv3'" {
        description
            "This augmentation is only valid for OSPFv3.";
    }
    description
        "Node MSD TLV is an optional TLV of OSPFv3 RI Opaque
        LSA (RFC7770) and has a type of 12.";

    uses node-msd-tlv;
}

augment "/rt:routing/"
    + "rt:control-plane-protocols/rt:control-plane-protocol/"
    + "ospf:ospf/ospf:database/"
    + "ospf:as-scope-lsa-type/ospf:as-scope-lsas/"
    + "ospf:as-scope-lsa/ospf:version/ospf:ospfv3/"
    + "ospf:ospfv3/ospf:body/ospf:router-information" {
when "../.../.../.../.../.../.../.../..."
    + "rt:type = 'ospf:ospfv3'" {
        description
            "This augmentation is only valid for OSPFv3.";
    }
    description
        "Node MSD TLV is an optional TLV of OSPFv3 RI Opaque
        LSA (RFC7770) and has a type of 12.";

    uses node-msd-tlv;
}

/* link MSD sub-tlv */
augment "/rt:routing/"
    + "rt:control-plane-protocols/rt:control-plane-protocol/"
    + "ospf:ospf/ospf:areas/ospf:area/"
    + "ospf:interfaces/ospf:interface/ospf:database/"
    + "ospf:link-scope-lsa-type/ospf:link-scope-lsas/"
    + "ospf:link-scope-lsa/ospf:version/ospf:ospfv2/"
    + "ospf:ospfv2/ospf:body/ospf:opaque/"
    + "ospf:extended-link-opaque/ospf:extended-link-tlv" {
when "../.../.../.../.../.../.../.../.../.../.../..."
    + "rt:type = 'ospf:ospfv2'" {
        description
            "This augmentation is only valid for OSPFv2.";
    }
    description

```

```

    "Link MSD sub-TLV is an optional sub-TLV of OSPFv2 extended
    link TLV as defined in RFC 7684 and has a type of 6.";

    uses link-msd-sub-tlv;
}

augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:areas/"
+ "ospf:area/ospf:database/"
+ "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
+ "ospf:area-scope-lsa/ospf:version/ospf:ospfv2/"
+ "ospf:ospfv2/ospf:body/ospf:opaque/"
+ "ospf:extended-link-opaque/ospf:extended-link-tlv" {
when "../.../.../.../.../.../.../.../.../.../.../..."
+ "rt:type = 'ospf:ospfv2'" {
    description
        "This augmentation is only valid for OSPFv2.";
}
description
    "Link MSD sub-TLV is an optional sub-TLV of OSPFv2 extended
    link TLV as defined in RFC 7684 and has a type of 6.";

    uses link-msd-sub-tlv;
}

augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:database/"
+ "ospf:as-scope-lsa-type/ospf:as-scope-lsas/"
+ "ospf:as-scope-lsa/ospf:version/ospf:ospfv2/"
+ "ospf:ospfv2/ospf:body/ospf:opaque/"
+ "ospf:extended-link-opaque/ospf:extended-link-tlv" {
when "../.../.../.../.../.../.../.../.../.../.../..."
+ "rt:type = 'ospf:ospfv2'" {
    description
        "This augmentation is only valid for OSPFv2.";
}
description
    "Link MSD sub-TLV is an optional sub-TLV of OSPFv2 extended
    link TLV as defined in RFC 7684 and has a type of 6.";

    uses link-msd-sub-tlv;
}

augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:areas/ospf:area/ospf:database/"

```

```
+ "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
+ "ospf:area-scope-lsa/ospf:version/ospf:ospfv3/"
+ "ospf:ospfv3/ospf:body/ospfv3-e-lsa:e-router"
+ "/ospfv3-e-lsa:e-router-tlvs" {
when "'ospf:.../.../.../.../.../.../.../.../'"
    + "rt:type' = 'ospf:ospfv3'" {
        description
            "This augmentation is only valid for OSPFv3
              E-Router LSAs";
    }
description
    "Augment OSPFv3 Area scope router-link TLV.";

uses link-msd-sub-tlv;
}
}
```

<CODE ENDS>

3. OSPF Segment Routing

This document defines a model for OSPF Segment Routing feature [RFC8665]. It is an augmentation of the OSPF base model.

The OSPF SR YANG module requires support for the base segment routing module [RFC9020], which defines the global segment routing configuration independent of any specific routing protocol configuration, and support of OSPF base model[I-D.ietf-ospf-yang] which defines basic OSPF configuration and state.

```

module: ietf-ospf-sr
augment /rt:routing/rt:control-plane-protocols
      /rt:control-plane-protocol/ospf:ospf:
  +--rw segment-routing
  |   +--rw enabled?      boolean
  |   +--rw bindings {mapping-server}?
  |   |   +--rw advertise
  |   |   |   +--rw policies* -> /rt:routing/sr:segment-routing
  |   |   |   |   /sr-mpls:sr-mpls/bindings
  |   |   |   |   /mapping-server/policy/name
  |   |   +--rw receive?      boolean
  +--rw protocol-srgb {sr-mpls:protocol-srgb}?
  |   +--rw srgb* [lower-bound upper-bound]
  |   +--rw lower-bound      uint32
  |   +--rw upper-bound      uint32
augment /rt:routing/rt:control-plane-protocols
      /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
      /ospf:interfaces/ospf:interface:

```

```

+--rw segment-routing
  +--rw adjacency-sid
    +--rw adj-sids* [value]
      |   +--rw value-type?  enumeration
      |   +--rw value        uint32
      |   +--rw protected?   boolean
      |   +--rw weight?      uint8
      +--rw advertise-adj-group-sid* [group-id]
        |   +--rw group-id    uint32
        +--rw advertise-protection? enumeration
augment /rt:routing/rt:control-plane-protocols
/rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
/ospf:interfaces/ospf:interface/ospf:fast-reroute:
+--rw ti-lfa {ti-lfa}?
  +--rw enable?    boolean
augment /rt:routing/rt:control-plane-protocols
/rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
/ospf:interfaces/ospf:interface/ospf:database
/ospf:link-scope-lsa-type/ospf:link-scope-lsas
/ospf:link-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
/ospf:body/ospf:opaque/ospf:extended-prefix-opaque:
+--ro extended-prefix-range-tlvs
  +--ro extended-prefix-range-tlv* []
    +--ro prefix-length?            uint8
    +--ro af?                       uint8
    +--ro range-size?               uint16
    +--ro extended-prefix-range-flags
      |   +--ro bits*    identityref
    +--ro prefix?                  inet:ip-prefix
    +--ro prefix-sid-sub-tlvs
      |   +--ro prefix-sid-sub-tlv* []
      |   |   +--ro prefix-sid-flags
      |   |   |   +--ro bits*    identityref
      |   |   +--ro mt-id?        uint8
      |   |   +--ro algorithm?    uint8
      |   |   +--ro sid?          uint32
    +--ro unknown-tlvs
      +--ro unknown-tlv* []
        +--ro type?    uint16
        +--ro length?  uint16
        +--ro value?   yang:hex-string
augment /rt:routing/rt:control-plane-protocols
/rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
/ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
/ospf:area-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
/ospf:body/ospf:opaque/ospf:extended-prefix-opaque:
+--ro extended-prefix-range-tlvs
  +--ro extended-prefix-range-tlv* []

```

```

+--ro prefix-length?                uint8
+--ro af?                            uint8
+--ro range-size?                    uint16
+--ro extended-prefix-range-flags
|   +--ro bits*    identityref
+--ro prefix?                        inet:ip-prefix
+--ro prefix-sid-sub-tlvs
|   +--ro prefix-sid-sub-tlv* []
|   |   +--ro prefix-sid-flags
|   |   |   +--ro bits*    identityref
|   |   +--ro mt-id?        uint8
|   |   +--ro algorithm?    uint8
|   |   +--ro sid?          uint32
+--ro unknown-tlvs
|   +--ro unknown-tlv* []
|   |   +--ro type?        uint16
|   |   +--ro length?      uint16
|   |   +--ro value?       yang:hex-string
augment /rt:routing/rt:control-plane-protocols
/rt:control-plane-protocol/ospf:ospf/ospf:database
/ospf:as-scope-lsa-type/ospf:as-scope-lsas
/ospf:as-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
/ospf:body/ospf:opaque/ospf:extended-prefix-opaque:
+--ro extended-prefix-range-tlvs
|   +--ro extended-prefix-range-tlv* []
|   |   +--ro prefix-length?                uint8
|   |   +--ro af?                            uint8
|   |   +--ro range-size?                    uint16
|   |   +--ro extended-prefix-range-flags
|   |   |   +--ro bits*    identityref
|   |   +--ro prefix?                        inet:ip-prefix
|   |   +--ro prefix-sid-sub-tlvs
|   |   |   +--ro prefix-sid-sub-tlv* []
|   |   |   |   +--ro prefix-sid-flags
|   |   |   |   |   +--ro bits*    identityref
|   |   |   +--ro mt-id?        uint8
|   |   |   +--ro algorithm?    uint8
|   |   |   +--ro sid?          uint32
|   |   +--ro unknown-tlvs
|   |   |   +--ro unknown-tlv* []
|   |   |   |   +--ro type?        uint16
|   |   |   |   +--ro length?      uint16
|   |   |   |   +--ro value?       yang:hex-string
augment /rt:routing/rt:control-plane-protocols
/rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
/ospf:interfaces/ospf:interface/ospf:database
/ospf:link-scope-lsa-type/ospf:link-scope-lsas
/ospf:link-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2

```

```

        /ospf:body/ospf:opaque/ospf:extended-prefix-opaque
        /ospf:extended-prefix-tlv:
+---ro prefix-sid-sub-tlvs
  +---ro prefix-sid-sub-tlv* []
    +---ro prefix-sid-flags
      | +---ro bits* identityref
    +---ro mt-id? uint8
    +---ro algorithm? uint8
    +---ro sid? uint32
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
  /ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
  /ospf:area-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
  /ospf:body/ospf:opaque/ospf:extended-prefix-opaque
  /ospf:extended-prefix-tlv:
+---ro prefix-sid-sub-tlvs
  +---ro prefix-sid-sub-tlv* []
    +---ro prefix-sid-flags
      | +---ro bits* identityref
    +---ro mt-id? uint8
    +---ro algorithm? uint8
    +---ro sid? uint32
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/ospf:ospf/ospf:database
  /ospf:as-scope-lsa-type/ospf:as-scope-lsas/ospf:as-scope-lsa
  /ospf:version/ospf:ospfv2/ospf:ospfv2/ospf:body/ospf:opaque
  /ospf:extended-prefix-opaque/ospf:extended-prefix-tlv:
+---ro prefix-sid-sub-tlvs
  +---ro prefix-sid-sub-tlv* []
    +---ro prefix-sid-flags
      | +---ro bits* identityref
    +---ro mt-id? uint8
    +---ro algorithm? uint8
    +---ro sid? uint32
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
  /ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
  /ospf:area-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
  /ospf:body/ospf:opaque/ospf:extended-link-opaque
  /ospf:extended-link-tlv:
+---ro adj-sid-sub-tlvs
  | +---ro adj-sid-sub-tlv* []
  | +---ro adj-sid-flags
  | | +---ro bits* identityref
  | +---ro mt-id? uint8
  | +---ro weight? uint8
  | +---ro sid? uint32
+---ro lan-adj-sid-sub-tlvs

```

```

    +---ro lan-adj-sid-sub-tlv* []
    +---ro lan-adj-sid-flags
    |   +---ro bits*      identityref
    +---ro mt-id?          uint8
    +---ro weight?         uint8
    +---ro neighbor-router-id? yang:dotted-quad
    +---ro sid?            uint32
augment /rt:routing/rt:control-plane-protocols
/rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
/ospf:interfaces/ospf:interface/ospf:database
/ospf:link-scope-lsa-type/ospf:link-scope-lsas
/ospf:link-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
/ospf:body/ospf:opaque/ospf:ri-opaque:
+---ro sr-algorithm-tlv
|   +---ro sr-algorithm*   uint8
+---ro sid-range-tlvs
|   +---ro sid-range-tlv* []
|   +---ro range-size?    uint24
|   +---ro sid-sub-tlv
|   |   +---ro sid?      uint32
+---ro local-block-tlvs
|   +---ro local-block-tlv* []
|   +---ro range-size?    uint24
|   +---ro sid-sub-tlv
|   |   +---ro sid?      uint32
+---ro srms-preference-tlv
    +---ro preference?    uint8
augment /rt:routing/rt:control-plane-protocols
/rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area
/ospf:database/ospf:area-scope-lsa-type/ospf:area-scope-lsas
/ospf:area-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
/ospf:body/ospf:opaque/ospf:ri-opaque:
+---ro sr-algorithm-tlv
|   +---ro sr-algorithm*   uint8
+---ro sid-range-tlvs
|   +---ro sid-range-tlv* []
|   +---ro range-size?    uint24
|   +---ro sid-sub-tlv
|   |   +---ro sid?      uint32
+---ro local-block-tlvs
|   +---ro local-block-tlv* []
|   +---ro range-size?    uint24
|   +---ro sid-sub-tlv
|   |   +---ro sid?      uint32
+---ro srms-preference-tlv
    +---ro preference?    uint8
augment /rt:routing/rt:control-plane-protocols
/rt:control-plane-protocol/ospf:ospf/database

```



```

        /ospf:as-scope-lsa-type/ospf:as-scope-lsas/ospf:as-scope-lsa
        /ospf:version/ospf:ospfv2/ospf:ospfv2/ospf:body/ospf:opaque
        /ospf:ri-opaque:
+--ro sr-algorithm-tlv
|   +--ro sr-algorithm*      uint8
+--ro sid-range-tlvs
|   +--ro sid-range-tlv* []
|       +--ro range-size?    uint24
|       +--ro sid-sub-tlv
|           +--ro sid?        uint32
+--ro local-block-tlvs
|   +--ro local-block-tlv* []
|       +--ro range-size?    uint24
|       +--ro sid-sub-tlv
|           +--ro sid?        uint32
+--ro srms-preference-tlv
    +--ro preference?        uint8

```

3.1. OSPF Segment Routing YANG Module

```

<CODE BEGINS> file "ietf-ospf-sr@2022-01-02.yang"
module ietf-ospf-sr {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-ospf-sr";

  prefix ospf-sr;

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

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

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

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

```

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

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

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

  Editor:     Derek Yeung
               <mailto:derek@arccus.com>
  Author:     Derek Yeung
               <mailto:derek@arccus.com>
  Author:     Yingzhen Qu
               <mailto:yingzhen.qu@futurewei.com>
  Author:     Acee Lindem
               <mailto:acee@cisco.com>
  Author:     Jeffrey Zhang
               <mailto:zzhang@juniper.net>
  Author:     Ing-Wher Chen
               <mailto:ingwherchen@mitre.org>
  Author:     Greg Hankins
               <mailto:greg.hankins@alcatel-lucent.com>";

description
  "This YANG module defines the generic configuration
  and operational state for OSPF Segment Routing, which is
  common across all of the vendor implementations. It is
  intended that the module will be extended by vendors to
  define vendor-specific OSPF Segment Routing configuration
  and operational parameters and policies.

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

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

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

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

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

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

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

reference "RFC XXXX";

```
revision 2022-01-02 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: A YANG Data Model for OSPF Segment Routing.";
}
```

```
feature ti-lfa {
  description
    "Topology-Independent Loop-Free Alternate (TI-LFA)
    computation using segment routing.";
}
```

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

```
identity np-bit {
  base prefix-sid-bit;
  description
    "No-PHP flag.";
}
```

```
identity m-bit {
  base prefix-sid-bit;
  description
```

```
        "Mapping server flag.";
    }

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

    identity v-bit {
        base prefix-sid-bit;
        description
            "Value/Index flag.";
    }

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

    identity extended-prefix-range-bit {
        description
            "Base identity for extended prefix range TLV bits.";
    }

    identity ia-bit {
        base extended-prefix-range-bit;
        description
            "Inter-Area flag. If set, advertisement is of inter-area type.";
    }

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

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

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

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

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

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

typedef uint24 {
    type uint32 {
        range "0 .. 16777215";
    }
    description
        "24-bit unsigned integer.";
}

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

grouping prefix-sid-sub-tlvs {
    description "Prefix Segment ID (SID) sub-TLVs.";
    container prefix-sid-sub-tlvs {
        description "Prefix SID sub-TLV.";
        list prefix-sid-sub-tlv {
            description "Prefix SID sub-TLV.";
        }
    }
}
```

```
    container prefix-sid-flags {
      leaf-list bits {
        type identityref {
          base prefix-sid-bit;
        }
        description
          "Prefix SID Sub-TLV flag bits list.";
      }
      description "Segment Identifier (SID) Flags.";
    }
    leaf mt-id {
      type uint8;
      description "Multi-topology ID.";
    }
    leaf algorithm {
      type uint8;
      description
        "The algorithm associated with the prefix-SID.";
    }
    leaf sid {
      type uint32;
      description "An index or label.";
    }
  }
}

grouping extended-prefix-range-tlvs {
  description "Extended prefix range TLV grouping.";

  container extended-prefix-range-tlvs {
    description "The list of range of prefixes.";
    list extended-prefix-range-tlv {
      description "The range of prefixes.";
      leaf prefix-length {
        type uint8;
        description "Length of prefix in bits.";
      }
      leaf af {
        type uint8;
        description "Address family for the prefix.";
      }
      leaf range-size {
        type uint16;
        description "The number of prefixes covered by the
          advertisement.";
      }
      container extended-prefix-range-flags {
```

```

        leaf-list bits {
            type identityref {
                base extended-prefix-range-bit;
            }
            description "Extended prefix range TLV flags list.";
        }
        description "Extended Prefix Range TLV flags.";
    }
    leaf prefix {
        type inet:ip-prefix;
        description "Address prefix.";
    }
    uses prefix-sid-sub-tlvs;
    uses ospf:unknown-tlvs;
}
}

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

grouping sid-range-tlvs {
    description "SID Range TLV grouping.";
    container sid-range-tlvs {
        description "List of SID range TLVs.";
        list sid-range-tlv {
            description "SID range TLV.";
            leaf range-size {
                type uint24;
                description "The SID range.";
            }
            uses sid-sub-tlv;
        }
    }
}

grouping local-block-tlvs {
    description "The SR local block TLV contains the

```

```

        range of labels reserved for local SIDs.";
    container local-block-tlvs {
        description "List of SRLB TLVs.";
        list local-block-tlv {
            description "SRLB TLV.";
            leaf range-size {
                type uint24;
                description "The SID range.";
            }
            uses sid-sub-tlv;
        }
    }
}

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

/* Configuration */
augment "/rt:routing/rt:control-plane-protocols"
    + "/rt:control-plane-protocol/ospf:ospf" {
    when "../rt:type = 'ospf:ospfv2' or "
    + "../rt:type = 'ospf:ospfv3'" {
        description
            "This augments the OSPF routing protocol when used.";
    }
    description
        "This augments the OSPF protocol configuration
        with segment routing.";
    uses sr-mpls:sr-control-plane;
    container protocol-srgb {
        if-feature sr-mpls:protocol-srgb;
        uses sr-cmn:srgb;
        description
            "Per-protocol SRGB.";
    }
}

```



```

augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/ospf:ospf/"
  + "ospf:areas/ospf:area/ospf:interfaces/ospf:interface" {
when "../.../.../rt:type = 'ospf:ospfv2' or "
  + "../.../.../rt:type = 'ospf:ospfv3'" {
  description
    "This augments the OSPF interface configuration
    when used.";
}
description
  "This augments the OSPF protocol interface
  configuration with segment routing.";

  uses sr-mpls:igp-interface;
}

augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/ospf:ospf/"
  + "ospf:areas/ospf:area/ospf:interfaces/ospf:interface/"
  + "ospf:fast-reroute" {
when "../.../.../rt:type = 'ospf:ospfv2' or "
  + "../.../.../rt:type = 'ospf:ospfv3'" {
  description
    "This augments the OSPF routing protocol when used.";
}
description
  "This augments the OSPF protocol IP-FRR with TI-LFA.";

  container ti-lfa {
    if-feature ti-lfa;
    leaf enable {
      type boolean;
      description
        "Enables TI-LFA computation.";
    }
    description
      "Topology Independent Loop Free Alternate
      (TI-LFA) support.";
  }
}

/* Database */
augment "/rt:routing/"
  + "rt:control-plane-protocols/rt:control-plane-protocol/"
  + "ospf:ospf/ospf:areas/ospf:area/"
  + "ospf:interfaces/ospf:interface/ospf:database/"
  + "ospf:link-scope-lsa-type/ospf:link-scope-lsas/"
  + "ospf:link-scope-lsa/ospf:version/ospf:ospfv2/"

```

```

    + "ospf:ospfv2/ospf:body/ospf:opaque/"
    + "ospf:extended-prefix-opaque" {
when "../.../.../.../.../.../.../.../.../.../.../..."
    + "rt:type = 'ospf:ospfv2'" {
    description
        "This augmentation is only valid for OSPFv2.";
    }
    description
        "SR specific TLVs for OSPFv2 extended prefix LSA
        in type 9 opaque LSA.";
    uses extended-prefix-range-tlvs;
}

augment "/rt:routing/"
    + "rt:control-plane-protocols/rt:control-plane-protocol/"
    + "ospf:ospf/ospf:areas/"
    + "ospf:area/ospf:database/"
    + "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
    + "ospf:area-scope-lsa/ospf:version/ospf:ospfv2/"
    + "ospf:ospfv2/ospf:body/ospf:opaque/"
    + "ospf:extended-prefix-opaque" {
when "../.../.../.../.../.../.../.../.../.../.../..."
    + "rt:type = 'ospf:ospfv2'" {
    description
        "This augmentation is only valid for OSPFv2.";
    }
    description
        "SR specific TLVs for OSPFv2 extended prefix LSA
        in type 10 opaque LSA.";
    uses extended-prefix-range-tlvs;
}

augment "/rt:routing/"
    + "rt:control-plane-protocols/rt:control-plane-protocol/"
    + "ospf:ospf/ospf:database/"
    + "ospf:as-scope-lsa-type/ospf:as-scope-lsas/"
    + "ospf:as-scope-lsa/ospf:version/ospf:ospfv2/"
    + "ospf:ospfv2/ospf:body/ospf:opaque/"
    + "ospf:extended-prefix-opaque" {
when "../.../.../.../.../.../.../.../.../.../.../..."
    + "rt:type = 'ospf:ospfv2'" {
    description
        "This augmentation is only valid for OSPFv2.";
    }
    description
        "SR specific TLVs for OSPFv2 extended prefix LSA

```

```

        in type 11 opaque LSA.";

    uses extended-prefix-range-tlvs;
}

augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:areas/ospf:area/"
+ "ospf:interfaces/ospf:interface/ospf:database/"
+ "ospf:link-scope-lsa-type/ospf:link-scope-lsas/"
+ "ospf:link-scope-lsa/ospf:version/ospf:ospfv2/"
+ "ospf:ospfv2/ospf:body/ospf:opaque/"
+ "ospf:extended-prefix-opaque/ospf:extended-prefix-tlv" {
when "../.../.../.../.../.../.../.../.../.../..."
+ "rt:type = 'ospf:ospfv2'" {
    description
        "This augmentation is only valid for OSPFv2.";
}
description
    "SR specific TLVs for OSPFv2 extended prefix TLV
    in type 9 opaque LSA.";
    uses prefix-sid-sub-tlvs;
}

augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:areas/"
+ "ospf:area/ospf:database/"
+ "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
+ "ospf:area-scope-lsa/ospf:version/ospf:ospfv2/"
+ "ospf:ospfv2/ospf:body/ospf:opaque/"
+ "ospf:extended-prefix-opaque/ospf:extended-prefix-tlv" {
when "../.../.../.../.../.../.../.../.../.../..."
+ "rt:type = 'ospf:ospfv2'" {
    description
        "This augmentation is only valid for OSPFv2.";
}
description
    "SR specific TLVs for OSPFv2 extended prefix TLV
    in type 10 opaque LSA.";
    uses prefix-sid-sub-tlvs;
}

augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:database/"
+ "ospf:as-scope-lsa-type/ospf:as-scope-lsas/"
+ "ospf:as-scope-lsa/ospf:version/ospf:ospfv2/"

```

```

    + "ospf:ospfv2/ospf:body/ospf:opaque/"
    + "ospf:extended-prefix-opaque/ospf:extended-prefix-tlv" {
when "../.../.../.../.../.../.../.../.../.../..."
    + "rt:type = 'ospf:ospfv2'" {
    description
        "This augmentation is only valid for OSPFv2.";
    }
    description
        "SR specific TLVs for OSPFv2 extended prefix TLV
        in type 11 opaque LSA.";
    uses prefix-sid-sub-tlvs;
}

augment "/rt:routing/"
    + "rt:control-plane-protocols/rt:control-plane-protocol/"
    + "ospf:ospf/ospf:areas/"
    + "ospf:area/ospf:database/"
    + "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
    + "ospf:area-scope-lsa/ospf:version/ospf:ospfv2/"
    + "ospf:ospfv2/ospf:body/ospf:opaque/"
    + "ospf:extended-link-opaque/ospf:extended-link-tlv" {
when "../.../.../.../.../.../.../.../.../.../..."
    + "rt:type = 'ospf:ospfv2'" {
    description
        "This augmentation is only valid for OSPFv2.";
    }
    description
        "SR specific TLVs for OSPFv2 extended link TLV
        in type 10 opaque LSA.";

    container adj-sid-sub-tlvs {
        description "Adjacency SID optional sub-TLVs.";
        list adj-sid-sub-tlv {
            description "List of Adjacency SID sub-TLVs.";
            container adj-sid-flags {
                leaf-list bits {
                    type identityref {
                        base adj-sid-bit;
                    }
                    description "Adj sid sub-tlv flags list.";
                }
                description "Adj-sid sub-tlv flags.";
            }
            leaf mt-id {
                type uint8;
                description "Multi-topology ID.";
            }
            leaf weight {

```

```

        type uint8;
        description "Weight used for load-balancing.";
    }
    leaf sid {
        type uint32;
        description "Segment Identifier (SID) index/label.";
    }
}

container lan-adj-sid-sub-tlvs {
    description "LAN Adjacency SID optional sub-TLVs.";
    list lan-adj-sid-sub-tlv {
        description "List of LAN adjacency SID sub-TLVs.";
        container lan-adj-sid-flags {
            leaf-list bits {
                type identityref {
                    base adj-sid-bit;
                }
                description "LAN adj sid sub-tlv flags list.";
            }
            description "LAN adj-sid sub-tlv flags.";
        }
        leaf mt-id {
            type uint8;
            description "Multi-topology ID.";
        }
        leaf weight {
            type uint8;
            description "Weight used for load-balancing.";
        }
        leaf neighbor-router-id {
            type yang:dotted-quad;
            description "Neighbor router ID.";
        }
        leaf sid {
            type uint32;
            description "Segment Identifier (SID) index/label.";
        }
    }
}

augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:areas/ospf:area/"
+ "ospf:interfaces/ospf:interface/ospf:database/"
+ "ospf:link-scope-lsa-type/ospf:link-scope-lsas/"

```

```

        + "ospf:link-scope-lsa/ospf:version/ospf:ospfv2/"
        + "ospf:ospfv2/ospf:body/ospf:opaque/ospf:ri-opaque" {
when "../.../.../.../.../.../.../.../.../.../.../..."
    + "rt:type = 'ospf:ospfv2'" {
        description
            "This augmentation is only valid for OSPFv2.";
    }

description
    "SR specific TLVs for OSPFv2 type 9 opaque LSA.";

uses sr-algorithm-tlv;
uses sid-range-tlvs;
uses local-block-tlvs;
uses srms-preference-tlv;
}

augment "/rt:routing/"
    + "rt:control-plane-protocols/rt:control-plane-protocol/"
    + "ospf:ospf/ospf:areas/"
    + "ospf:area/ospf:database/"
    + "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
    + "ospf:area-scope-lsa/ospf:version/ospf:ospfv2/"
    + "ospf:ospfv2/ospf:body/ospf:opaque/ospf:ri-opaque" {
when "../.../.../.../.../.../.../.../.../.../..."
    + "rt:type = 'ospf:ospfv2'" {
        description
            "This augmentation is only valid for OSPFv2.";
    }

description
    "SR specific TLVs for OSPFv2 type 10 opaque LSA.";

uses sr-algorithm-tlv;
uses sid-range-tlvs;
uses local-block-tlvs;
uses srms-preference-tlv;
}

augment "/rt:routing/"
    + "rt:control-plane-protocols/rt:control-plane-protocol/"
    + "ospf:ospf/ospf:database/"
    + "ospf:as-scope-lsa-type/ospf:as-scope-lsas/"
    + "ospf:as-scope-lsa/ospf:version/ospf:ospfv2/"
    + "ospf:ospfv2/ospf:body/ospf:opaque/ospf:ri-opaque" {
when "../.../.../.../.../.../.../.../.../..."
    + "rt:type = 'ospf:ospfv2'" {
        description

```

```

        "This augmentation is only valid for OSPFv2.";
    }
    description
        "SR specific TLVs for OSPFv2 type 11 opaque LSA.";

    uses sr-algorithm-tlv;
    uses sid-range-tlvs;
    uses local-block-tlvs;
    uses srms-preference-tlv;
}
}
<CODE ENDS>

```

4. Security Considerations

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

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

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

/ospf:ospf/segment-routing/enabled - Modification to the enablement for SR could result in a Denial-of-Service (Dos) attack. If an attacker disables SR, it will cause traffic disruption.

/ospf:ospf/segment-routing/bindings - Modification to the local bindings could result in a Denial-of-Service (Dos) attack.

/ospf:ospf/protocol-srgb - Modification of the protocol SRGB could be used to mount a DoS attack. For example, if the protocol SRBG size is reduced to a very small value, a lot of existing segments could no longer be installed leading to a traffic disruption.

/ospf:interfaces/ospf:interface/segment-routing - Modification of the Adjacency Segment Identifier (Adj-SID) could be used to mount a DoS attack. Change of an Adj-SID could be used to redirect traffic.

/ospf:interfaces/ospf:interface/ospf:fast-reroute/ti-lfa - Modification of the TI-LFA enablement could lead to traffic disruption.

Some of the readable data nodes in the modules may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes.

Both module ietf-ospf-sr and ietf-ospf-msd augment base OSPF module data base with various TLVs. Knowledge of these data nodes can be used to attack other routers in the OSPF domain.

5. Acknowledgements

The authors wish to thank Yi Yang, Alexander Clemm, Gaurav Gupta, Ladislav Lhotka, Stephane Litkowski, Greg Hankins, Manish Gupta and Alan Davey for their thorough reviews and helpful comments.

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

Author affiliation with The MITRE Corporation is provided for identification purposes only, and is not intended to convey or imply MITRE's concurrence with, or support for, the positions, opinions or viewpoints expressed. MITRE has approved this document for Public Release, Distribution Unlimited, with Public Release Case Number 18-3281.

6. IANA Considerations

This document registers a URI in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested to be made:

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

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

This document registers a YANG module in the YANG Module Names registry [RFC6020].

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

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

7. References

7.1. Normative References

- [I-D.ietf-ospf-yang] Yeung, D., Qu, Y., Zhang, J., Chen, I., and A. Lindem, "YANG Data Model for OSPF Protocol", Work in Progress, Internet-Draft, draft-ietf-ospf-yang-29, 17 October 2019, <<https://www.ietf.org/archive/id/draft-ietf-ospf-yang-29.txt>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <<https://www.rfc-editor.org/info/rfc2328>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC4750] Joyal, D., Ed., Galecki, P., Ed., Giacalone, S., Ed., Coltun, R., and F. Baker, "OSPF Version 2 Management Information Base", RFC 4750, DOI 10.17487/RFC4750, December 2006, <<https://www.rfc-editor.org/info/rfc4750>>.
- [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <<https://www.rfc-editor.org/info/rfc5340>>.

- [RFC5643] Joyal, D., Ed. and V. Manral, Ed., "Management Information Base for OSPFv3", RFC 5643, DOI 10.17487/RFC5643, August 2009, <<https://www.rfc-editor.org/info/rfc5643>>.
- [RFC5838] Lindem, A., Ed., Mirtorabi, S., Roy, A., Barnes, M., and R. Aggarwal, "Support of Address Families in OSPFv3", RFC 5838, DOI 10.17487/RFC5838, April 2010, <<https://www.rfc-editor.org/info/rfc5838>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 7223, DOI 10.17487/RFC7223, May 2014, <<https://www.rfc-editor.org/info/rfc7223>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8476] Tantsura, J., Chunduri, U., Aldrin, S., and P. Psenak, "Signaling Maximum SID Depth (MSD) Using OSPF", RFC 8476, DOI 10.17487/RFC8476, December 2018, <<https://www.rfc-editor.org/info/rfc8476>>.

- [RFC8665] Psenak, P., Ed., Previdi, S., Ed., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", RFC 8665, DOI 10.17487/RFC8665, December 2019, <<https://www.rfc-editor.org/info/rfc8665>>.
- [RFC9020] Litkowski, S., Qu, Y., Lindem, A., Sarkar, P., and J. Tantsura, "YANG Data Model for Segment Routing", RFC 9020, DOI 10.17487/RFC9020, May 2021, <<https://www.rfc-editor.org/info/rfc9020>>.

7.2. Informative References

- [RFC8022] Lhotka, L. and A. Lindem, "A YANG Data Model for Routing Management", RFC 8022, DOI 10.17487/RFC8022, November 2016, <<https://www.rfc-editor.org/info/rfc8022>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.

Appendix A. Contributors' Addreses

Dean Bogdanovic
Volta Networks, Inc.

EMail: dean@voltanet.io

Kiran Koushik Agrahara Sreenivasa
Cisco Systems
12515 Research Blvd, Bldg 4
Austin, TX 78681
USA

EMail: kkoushik@cisco.com

Authors' Addresses

Derek Yeung
Arrcus

Email: derek@arrcus.com

Yingzhen Qu
Futurewei
2330 Central Expressway
Santa Clara, CA 95050
United States of America

Email: yingzhen.qu@futurewei.com

Jeffrey Zhang
Juniper Networks
10 Technology Park Drive
Westford, MA 01886
United States of America

Email: zzhang@juniper.net

Ing-Wher Chen
The MITRE Corporation

Email: ingwherchen@mitre.org

Acee Lindem
Cisco Systems
301 Midenhall Way
Cary, NC 27513

Email: acee@cisco.com

Link State Routing
Internet-Draft
Intended status: Standards Track
Expires: January 9, 2020

K. Talaulikar
P. Psenak
Cisco Systems, Inc.
July 8, 2019

OSPF Strict-Mode for BFD
draft-ketant-lsr-ospf-bfd-strict-mode-02

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 January 9, 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	7
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 indicate to enable discovery of the IPv4 addresses for BFD session setup.

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

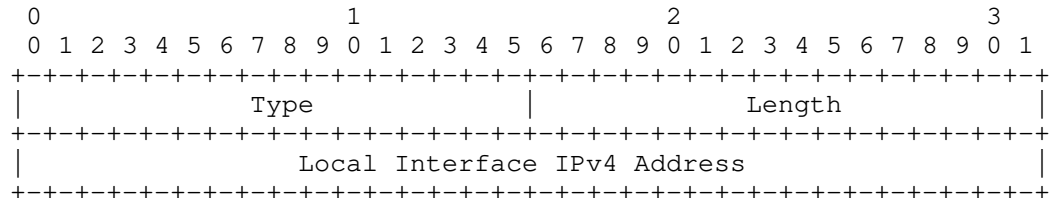
2. LLS B-bit Flag

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.

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

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

Email: ppsenak@cisco.com

Link State Routing
Internet-Draft
Intended status: Standards Track
Expires: January 1, 2021

K. Talaulikar
P. Psenak
Cisco Systems
June 30, 2020

Advertising L2 Bundle Member Link Attributes in OSPF
draft-ketant-lsr-ospf-l2bundles-02

Abstract

There are deployments where the Layer 3 interface on which OSPF operates is a Layer 2 interface bundle. Existing OSPF advertisements only support advertising link attributes of the Layer 3 interface. If entities external to OSPF wish to control traffic flows on the individual physical links which comprise the Layer 2 interface bundle link attribute information about the bundle members is required.

This document introduces the ability for OSPF to advertise the link attributes of layer 2 (L2) Bundle members.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 1, 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. L2 Bundle Member Attributes	3
3. IANA Considerations	6
4. Security Considerations	7
5. Acknowledgements	7
6. References	7
6.1. Normative References	7
6.2. Informational References	8
Authors' Addresses	9

1. Introduction

There are deployments where the Layer 3 interface on which an OSPF adjacency is established is a Layer 2 interface bundle, for instance a Link Aggregation Group (LAG) [IEEE802.1AX] . This reduces the number of adjacencies which need to be maintained by the routing protocol in cases where there are parallel links between the neighbors. Entities external to OSPF such as Path Computation Elements (PCE) [RFC4655] may wish to control traffic flows on individual members of the underlying Layer 2 bundle. In order to do so link attribute information about individual bundle members is required. The protocol extensions defined in this document provide the means to advertise this information.

This document introduces new sub-TLVs to advertise link attribute information for each of the L2 Bundle members which comprise the Layer 3 interface on which OSPF operates. Similar capabilities were introduced in IS-IS via [RFC8668].

[RFC8665] and [RFC8666] introduced the adjacency segment identifier (Adj-SID) link attribute for OSPFv2 and OSPFv3 respectively which can be used as an instruction to forwarding to send traffic over a specific link [RFC8402]. This document enables the advertisement of the Adj-SIDs using the same Adjacency SID sub-TLV at the granularity level of each L2 Bundle member link so that traffic may be steered over that specific member link.

Note that the new advertisements at the L2 Bundle member link level in this document are intended to be provided to external (to OSPF) entities and does not alter or change OSPF route computation process.

The following items are intentionally not defined and/or are outside the scope of this document:

- o What link attributes will be advertised. This is determined by the needs of the external entities.
- o A minimum or default set of link attributes.
- o How these attributes are configured
- o How the advertisements are used
- o What impact the use of these advertisements may have on traffic flow in the network
- o How the advertisements are passed to external entities

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. L2 Bundle Member Attributes

A new L2 Bundle Member Attributes sub-TLV is introduced to advertise L2 Bundle member attributes in both OSPFv2 and OSPFv3. In case of OSPFv2, this sub-TLV is an optional sub-TLV of the OSPFv2 Extended Link TLV that is used to describe link attributes via the OSPFv2 Extended Link Opaque LSA [RFC7684]. In case of OSPFv3, this sub-TLV is an optional sub-TLV of the Router Link TLV of the OSPFv3 E-Router-LSA [RFC8362].

When the OSPF adjacency is associated with L2 Bundle interface, this sub-TLV is used to advertise the underlying L2 Bundle member links along with their individual link attributes. Inclusion of this information implies that the identified link is a member of the L2 bundle associated with OSPF L3 link and that the member link is operationally up. Therefore advertisements of member links MUST NOT be done when the member link becomes operationally down or it is no longer a member of the identified L2 Bundle.

The L2 Bundle Member Attributes sub-TLV has the following format:

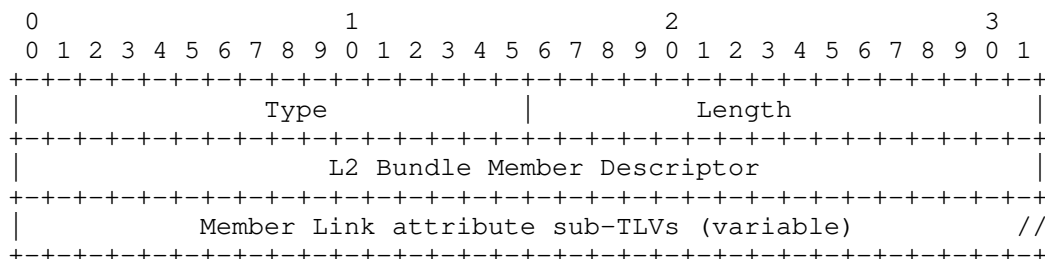


Figure 1: L2 Bundle Member Attributes sub-TLV Format

Where:

Type: TBD1 for OSPFv2 and TBD2 for OSPFv3

Length: Variable.

L2 Bundle Member Descriptor: A 4 octet Link Local Identifier as described in [RFC4202] and used in [RFC8510] for the member link.

Link attributes for L2 Bundle Member Links are advertised as sub-TLVs of the L2 Bundle Member Attribute sub-TLV.

In the case of OSPFv2, the L2 Bundle Member Attributes sub-TLV shares the sub-TLV space of the Extended Link TLV and the sub-TLVs of the Extended Link TLV MAY be used to describe the attributes of the member link. The Figure 2 below lists sub-TLVs and their applicability for L2 Bundle member links. The sub-TLVs that are not applicable MUST NOT be used as sub-TLVs for the L2 Bundle Member Attributes sub-TLV. Specifications that introduce new sub-TLVs of the Extended Link TLV MUST indicate their applicability for the L2 Bundle Member Attributes sub-TLV. An implementation MUST ignore any sub-TLVs received that are not applicable in the context of the L2 Bundle Member Attribute sub-TLV.

Y	-	applicable
N	-	not-applicable
1	SID/Label	(N)
2	Adj-SID	(Y)
3	LAN Adj-SID/Label	(Y)
4	Network-to-Router Metric	(N)
5	RTM Capability	(N)
6	OSPFv2 Link MSD	(N)
7	Graceful-Link-Shutdown	(N)
8	Remote IPv4 Address	(N)
9	Local/Remote Interface ID	(N)
10	Application Specific Link Attributes	(Y)
11	Shared Risk Link Group	(Y)
12	Unidirectional Link Delay	(Y)
13	Min/Max Unidirectional Link Delay	(Y)
14	Unidirectional Delay Variation	(Y)
15	Unidirectional Link Loss	(Y)
16	Unidirectional Residual Bandwidth	(Y)
17	Unidirectional Available Bandwidth	(Y)
18	Unidirectional Utilized Bandwidth	(Y)
19	Administrative Group	(Y)
20	Extended Administrative Group	(Y)
21	Maximum Link Bandwidth	(Y)
22	Traffic Engineering Metric	(Y)
TBD1	L2 Bundle Member Attributes	(N)

Figure 2: Applicability of OSPFv2 Link Attribute sub-TLVs for L2 Bundle Members

In the case of OSPFv3, the L2 Bundle Member Attributes sub-TLV shares the sub-TLV space of the Router Link TLV and the sub-TLVs of the Router Link TLV MAY be used to describe the attributes of the member link. The Figure 3 below lists sub-TLVs that are applicable for Router Link TLV and their applicability for L2 Bundle member links. The sub-TLVs that are not applicable MUST NOT be used as sub-TLVs for the L2 Bundle Member Attributes sub-TLV. Specifications that introduce new sub-TLVs of the Router Link TLV MUST indicate their applicability for the L2 Bundle Member Attributes sub-TLV. An implementation MUST ignore any sub-TLVs received that are not applicable in the context of the L2 Bundle Member Attribute sub-TLV.

Y - applicable
N - not-applicable

5	Adj-SID (Y)
6	LAN Adj-SID (Y)
7	SID/Label (N)
8	Graceful-Link-Shutdown (N)
9	OSPFv3 Link MSD (N)
10	Application Specific Link Attributes (Y)
11	Shared Risk Link Group (Y)
12	Unidirectional Link Delay (Y)
13	Min/Max Unidirectional Link Delay (Y)
14	Unidirectional Delay Variation (Y)
15	Unidirectional Link Loss (Y)
16	Unidirectional Residual Bandwidth (Y)
17	Unidirectional Available Bandwidth (Y)
18	Unidirectional Utilized Bandwidth (Y)
19	Administrative Group (Y)
20	Extended Administrative Group (Y)
21	Traffic Engineering Metric (Y)
22	Maximum Link Bandwidth (Y)
23	Local Interface IPv6 Address (N)
24	Remote Interface IPv6 Address (N)
TBD2	L2 Bundle Member Attributes (N)

Figure 3: Applicability of OSPFv3 Link Attribute sub-TLVs for L2 Bundle Members

3. IANA Considerations

This document adds new sub-TLVs to the OSPFv2 and OSPFv3 registry.

The following sub-TLV is added to the OSPFv2 Extended Link TLV sub-TLVs registry under the OSPFv2 Parameters IANA registry:

Value: TBD1

Name: L2 Bundle Member Attributes

The following sub-TLV is added to the OSPFv3 Extended LSA sub-TLVs registry under the OSPFv3 Parameters IANA registry:

Value: TBD2

Name: L2 Bundle Member Attributes

4. Security Considerations

The OSPF protocol has supported the advertisement of link attribute information, including link identifiers, for many years. The advertisements defined in this document are identical to existing advertisements defined in [RFC3630], [RFC4203], [RFC5329], [RFC7471], [RFC8665] and [RFC8666] - but are associated with L2 links which are part of a bundle interface on which the OSPF protocol operates. There are therefore no new security issues introduced by the extensions in this document.

As always, if the protocol is used in an environment where unauthorized access to the physical links on which OSPF packets are sent occurs then attacks are possible. The use of authentication as defined in [RFC5709], [RFC7474], [RFC4552] and [RFC7166] is recommended to prevent such attacks.

5. Acknowledgements

This document leverages the similar work done for IS-IS and the authors of this document would like to acknowledge the contributions of the authors of [RFC8668].

6. References

6.1. Normative References

- [IEEE802.1AX]
Institute of Electrical and Electronics Engineers, "IEEE Standard for Local and Metropolitan Area Networks - Link Aggregation.", Nov 2008.
- [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>>.
- [RFC4202] Kompella, K., Ed. and Y. Rekhter, Ed., "Routing Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4202, DOI 10.17487/RFC4202, October 2005, <<https://www.rfc-editor.org/info/rfc4202>>.
- [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>>.
- [RFC8665] Psenak, P., Ed., Previdi, S., Ed., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", RFC 8665, DOI 10.17487/RFC8665, December 2019, <<https://www.rfc-editor.org/info/rfc8665>>.
- [RFC8666] Psenak, P., Ed. and S. Previdi, Ed., "OSPFv3 Extensions for Segment Routing", RFC 8666, DOI 10.17487/RFC8666, December 2019, <<https://www.rfc-editor.org/info/rfc8666>>.

6.2. Informational References

- [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>>.
- [RFC4203] Kompella, K., Ed. and Y. Rekhter, Ed., "OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4203, DOI 10.17487/RFC4203, October 2005, <<https://www.rfc-editor.org/info/rfc4203>>.
- [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>>.
- [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>>.
- [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>>.

- [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>>.
- [RFC7166] Bhatia, M., Manral, V., and A. Lindem, "Supporting Authentication Trailer for OSPFv3", RFC 7166, DOI 10.17487/RFC7166, March 2014, <<https://www.rfc-editor.org/info/rfc7166>>.
- [RFC7471] Giacalone, S., Ward, D., Drake, J., Atlas, A., and S. Previdi, "OSPF Traffic Engineering (TE) Metric Extensions", RFC 7471, DOI 10.17487/RFC7471, March 2015, <<https://www.rfc-editor.org/info/rfc7471>>.
- [RFC7474] Bhatia, M., Hartman, S., Zhang, D., and A. Lindem, Ed., "Security Extension for OSPFv2 When Using Manual Key Management", RFC 7474, DOI 10.17487/RFC7474, April 2015, <<https://www.rfc-editor.org/info/rfc7474>>.
- [RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.
- [RFC8510] Psenak, P., Ed., Talaulikar, K., Henderickx, W., and P. Pillay-Esnault, "OSPF Link-Local Signaling (LLS) Extensions for Local Interface ID Advertisement", RFC 8510, DOI 10.17487/RFC8510, January 2019, <<https://www.rfc-editor.org/info/rfc8510>>.
- [RFC8668] Ginsberg, L., Ed., Bashandy, A., Filsfils, C., Nanduri, M., and E. Aries, "Advertising Layer 2 Bundle Member Link Attributes in IS-IS", RFC 8668, DOI 10.17487/RFC8668, December 2019, <<https://www.rfc-editor.org/info/rfc8668>>.

Authors' Addresses

Ketan Talaulikar
Cisco Systems
India

Email: ketant@cisco.com

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

Email: ppsenak@cisco.com

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

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: December 7, 2019

T. Li
Arista Networks
L. Ginsberg
P. Wells
Cisco Systems
June 5, 2019

Hierarchical IS-IS
draft-li-lsr-isis-hierarchical-isis-01

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 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	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. Level n LAN IS to IS hello PDU (Ln-LAN-HELLO-PDU)	4
3.2. Level n Point-to-point IS to IS hello PDU (Ln-P2P-HELLO-PDU)	4
4. IS-IS Area Identifier TLV	5
5. New Flooding Scopes	5
6. Inheritance of TLVs	6
7. Relationship between levels	7
8. Acknowledgements	7
9. IANA Considerations	7
9.1. PDU Type	7
9.2. New PDUs	7
9.3. New TLVs	7
9.4. New Flooding Scopes	8
10. Security Considerations	8
11. Normative References	9
Authors' Addresses	9

1. Introduction

The IS-IS routing protocol IS-IS [ISO10589] currently supports a two level hierarchy of abstraction. The fundamental unit of abstraction is the 'area', which is a (hopefully) connected set of systems running IS-IS at the same level. Level 1, the lowest level, is abstracted by routers that participate in both Level 1 and Level 2.

Practical considerations, such as the size of an area's link state database, cause network designers to restrict the number of routers in any given area. Concurrently, the dominance of scale-out architectures based around small routers has created a situation where the scalability limits of the protocol are going to become critical in the foreseeable future.

The goal of this document is to enable additional hierarchy within IS-IS. Each additional level of hierarchy has a multiplicative effect on scale, so the addition of six levels should be a

significant improvement. While all six levels may not be needed in the short term, it is apparent that the original designers of IS-IS reserved enough space for these levels, and defining six additional levels is only slightly harder than adding a single level, so it makes sense to expand the design for the future.

The modifications described herein are designed to be fully backward compatible and have no effect on existing networks. The modifications are also designed to have no effect whatsoever on networks that only use Level 1 and/or Level 2.

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

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. Level n LAN IS to IS hello PDU (Ln-LAN-HELLO-PDU)

The 'Level n LAN IS to IS hello PDU' (Ln-LAN-HELLO-PDU) is identical in format to the 'Level 2 LAN IS to IS hello PDU' (Section 9.6), except that the PDU Types are defined as follows:

Level 3 (L3-LAN-HELLO-PDU): AA3

Level 4 (L4-LAN-HELLO-PDU): AA4

Level 5 (L5-LAN-HELLO-PDU): AA5

Level 6 (L6-LAN-HELLO-PDU): AA6

Level 7 (L7-LAN-HELLO-PDU): AA7

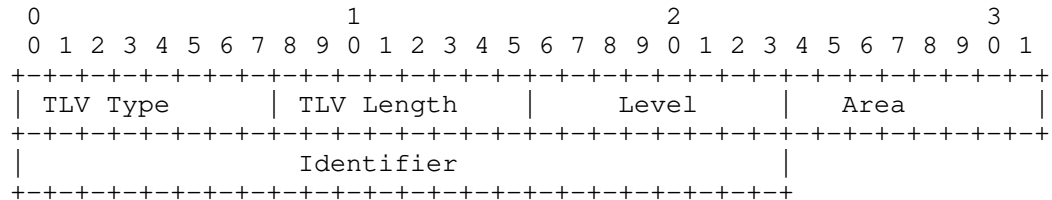
Level 8 (L8-LAN-HELLO-PDU): AA8

3.2. Level n Point-to-point IS to IS hello PDU (Ln-P2P-HELLO-PDU)

The 'Point-to-point IS to IS hello PDU' (Section 9.7) is used on Level 1 and Level 2 circuits. Legacy systems will not expect the circuit type field to indicate other levels, so a new PDU is used if the circuit supports other levels. The additional PDU is the 'Level n Point-to-point IS to IS hello PDU' (Ln-P2P-HELLO-PDU) and has PDU Type TTT with the same format. Both PDUs may be used on the same circuit.

4. IS-IS Area Identifier TLV

The Area Identifier TLV is added to IS-IS to allow nodes to indicate which areas they participate in. Area Identifiers are locally administered 32 bit numbers. The format of the TLV is:



TLV Type: ZZZ

TLV Length: 7

Level: The level number of the area.

Area Identifier: The identifier associated with the area.

The Area Identifier TLV may appear in IIHs or in LSPs. When the Area Identifier TLV appears in a PDU, it indicates that the system is participating in the specified area at the indicated level. When the Area Identifier TLV appears in a IIH, the receiving system MUST NOT form an adjacency unless an Area Identifier TLV corresponds to the receiver's own Area Identifier for the given level.

5. New Flooding Scopes

For levels 3-8, all link state information, PSNPs, and CSNPs are relayed in conformance with RFC 7356 [RFC7356]. Additional flooding scopes are defined for each new level, for both circuit flooding scope and level flooding scope. Level flooding scopes are defined for both Standard and Extended TLV formats. The list of additional flooding scopes is:

Value	Description	FS LSP ID Format/ TLV Format
6	Level 3 Circuit Flooding Scope	Extended/Standard
7	Level 4 Circuit Flooding Scope	Extended/Standard
8	Level 5 Circuit Flooding Scope	Extended/Standard
9	Level 6 Circuit Flooding Scope	Extended/Standard
10	Level 7 Circuit Flooding Scope	Extended/Standard
11	Level 8 Circuit Flooding Scope	Extended/Standard
12	Level 3 Flooding Scope	Extended/Standard
13	Level 4 Flooding Scope	Extended/Standard
14	Level 5 Flooding Scope	Extended/Standard
15	Level 6 Flooding Scope	Extended/Standard
16	Level 7 Flooding Scope	Extended/Standard
17	Level 8 Flooding Scope	Extended/Standard
18	Level 3 Flooding Scope	Standard/Standard
19	Level 4 Flooding Scope	Standard/Standard
20	Level 5 Flooding Scope	Standard/Standard
21	Level 6 Flooding Scope	Standard/Standard
22	Level 7 Flooding Scope	Standard/Standard
23	Level 8 Flooding Scope	Standard/Standard
70	Level 3 Circuit Flooding Scope	Extended/Extended
71	Level 4 Circuit Flooding Scope	Extended/Extended
72	Level 5 Circuit Flooding Scope	Extended/Extended
73	Level 6 Circuit Flooding Scope	Extended/Extended
74	Level 7 Circuit Flooding Scope	Extended/Extended
75	Level 8 Circuit Flooding Scope	Extended/Extended
76	Level 3 Flooding Scope	Extended/Extended
77	Level 4 Flooding Scope	Extended/Extended
78	Level 5 Flooding Scope	Extended/Extended
79	Level 6 Flooding Scope	Extended/Extended
80	Level 7 Flooding Scope	Extended/Extended
81	Level 8 Flooding Scope	Extended/Extended

6. 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.

7. Relationship between levels

The relationship between Level n and Level $n-1$ is analogous to the relationship between Level 2 and Level 1.

8. Acknowledgements

The author would like to thank Dinesh Dutt for inspiring this document. The author would also like to thank Les Ginsberg and Paul Wells for their helpful comments.

9. IANA Considerations

This document makes many requests to IANA, as follows:

9.1. 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.

9.2. New PDUs

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

L3-LAN-HELLO-PDU: AA3

L4-LAN-HELLO-PDU: AA4

L5-LAN-HELLO-PDU: AA5

L6-LAN-HELLO-PDU: AA6

L7-LAN-HELLO-PDU: AA7

L8-LAN-HELLO-PDU: AA8

Ln-P2P-HELLO-PDU: TTT

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

9.3. New TLVs

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

Area Identifier: ZZZ

9.4. New Flooding Scopes

IANA is requested to allocate the following values from the IS-IS Flooding Scope Identifier Registry.

Value	Description	FS LSP ID Format/ TLV Format	IIH Announce Lx-P2P Lx-LAN
6	Level 3 Circuit Flooding Scope	Extended/Standard	Y Y
7	Level 4 Circuit Flooding Scope	Extended/Standard	Y Y
8	Level 5 Circuit Flooding Scope	Extended/Standard	Y Y
9	Level 6 Circuit Flooding Scope	Extended/Standard	Y Y
10	Level 7 Circuit Flooding Scope	Extended/Standard	Y Y
11	Level 8 Circuit Flooding Scope	Extended/Standard	Y Y
12	Level 3 Flooding Scope	Extended/Standard	Y Y
13	Level 4 Flooding Scope	Extended/Standard	Y Y
14	Level 5 Flooding Scope	Extended/Standard	Y Y
15	Level 6 Flooding Scope	Extended/Standard	Y Y
16	Level 7 Flooding Scope	Extended/Standard	Y Y
17	Level 8 Flooding Scope	Extended/Standard	Y Y
18	Level 3 Flooding Scope	Standard/Standard	Y Y
19	Level 4 Flooding Scope	Standard/Standard	Y Y
20	Level 5 Flooding Scope	Standard/Standard	Y Y
21	Level 6 Flooding Scope	Standard/Standard	Y Y
22	Level 7 Flooding Scope	Standard/Standard	Y Y
23	Level 8 Flooding Scope	Standard/Standard	Y Y
70	Level 3 Circuit Flooding Scope	Extended/Extended	Y Y
71	Level 4 Circuit Flooding Scope	Extended/Extended	Y Y
72	Level 5 Circuit Flooding Scope	Extended/Extended	Y Y
73	Level 6 Circuit Flooding Scope	Extended/Extended	Y Y
74	Level 7 Circuit Flooding Scope	Extended/Extended	Y Y
75	Level 8 Circuit Flooding Scope	Extended/Extended	Y Y
76	Level 3 Flooding Scope	Extended/Extended	Y Y
77	Level 4 Flooding Scope	Extended/Extended	Y Y
78	Level 5 Flooding Scope	Extended/Extended	Y Y
79	Level 6 Flooding Scope	Extended/Extended	Y Y
80	Level 7 Flooding Scope	Extended/Extended	Y Y
81	Level 8 Flooding Scope	Extended/Extended	Y Y

10. 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.

11. 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>>.
- [RFC7356] Ginsberg, L., Previdi, S., and Y. Yang, "IS-IS Flooding
Scope Link State PDUs (LSPs)", RFC 7356,
DOI 10.17487/RFC7356, September 2014,
<<https://www.rfc-editor.org/info/rfc7356>>.

Authors' Addresses

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

Email: tony.li@tony.li

Les Ginsberg
Cisco Systems
United States of America

Email: ginsberg@cisco.com

Paul Wells
Cisco Systems
United States of America

Email: pauwells@cisco.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 9, 2020

Z. Li
Z. Hu
D. Cheng
Huawei Technologies
K. Talaulikar
P. Psenak
Cisco Systems
July 8, 2019

OSPFv3 Extensions for SRv6
draft-li-ospf-ospfv3-srv6-extensions-04

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 OSPFv3 extensions required to support Segment Routing over an IPv6 data plane (SRv6).

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on January 9, 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. SRv6-Capabilities TLV	3
3. Advertisement of Supported Algorithms	5
4. Advertisement of SRH Operation Limits	5
5. Advertisement of SRv6 Locator and End SIDs	5
6. SRv6 Locator LSA	6
6.1. SRv6 Locator TLV	8
7. Advertisement of SRv6 End SIDs	10
8. Advertisement of SRv6 SIDs Associated with Adjacencies	11
8.1. SRv6 End.X SID Sub-TLV	12
8.2. SRv6 LAN End.X SID Sub-TLV	14
9. SRv6 SID Structure sub-TLV	15
10. Security Considerations	16
11. IANA Considerations	16
11.1. OSPF Router Information TLVs	17
11.2. OSPFv3 LSA Function Codes	17
11.3. OSPFv3 Extended-LSA sub-TLVs	17
11.4. OSPFv3 Locator LSA TLVs	17
11.5. OSPFv3 Locator LSA sub-TLVs	18
12. Acknowledgements	18
13. References	19
13.1. Normative References	19
13.2. Informative References	20
Authors' Addresses	21

1. Introduction

Segment Routing (SR) architecture [RFC8402] specifies how a node can steer a packet through an ordered list of instructions, called segments. These segments are identified through Segment Identifiers (SIDs).

Segment Routing can be instantiated on the IPv6 data plane through the use of the Segment Routing Header (SRH) defined in [I-D.ietf-6man-segment-routing-header]. SRv6 refers to this SR instantiation on the IPv6 dataplane. The network programming paradigm for SRv6 is specified in [I-D.ietf-spring-srv6-network-programming] which describes several well-known functions that can be bound to SRv6 SIDs.

This document specifies extensions to OSPFv3 in order to support SRv6 as defined in [I-D.ietf-spring-srv6-network-programming] by signaling the SRv6 capabilities of the node and certain SRv6 SIDs with their endpoint behaviors (e.g. End, End.X, etc.) that are instantiated on the SRv6 capable router.

At a high level, the extensions to OSPFv3 comprise of the following:

1. SRv6 Capabilities TLV to advertise the support for SRv6 features and SRH operations supported by the router
 2. SRv6 Locator TLV to advertise the SRv6 Locator - a form of summary address for the algorithm specific SIDs associated with the router
 3. TLVs and sub-TLVs to advertise the SRv6 SIDs instantiated on the router along with their endpoint behaviors
2. SRv6-Capabilities TLV

When Segment Routing (SR) is instantiated using the IPv6 data plane (SRv6), the list of segments is expressed using the segment routing header (SRH) as defined in [I-D.ietf-6man-segment-routing-header].

A router that supports SRv6 MUST be able to process the SRH as described in [I-D.ietf-6man-segment-routing-header], as well as apply endpoint behaviors as described in [I-D.ietf-spring-srv6-network-programming].

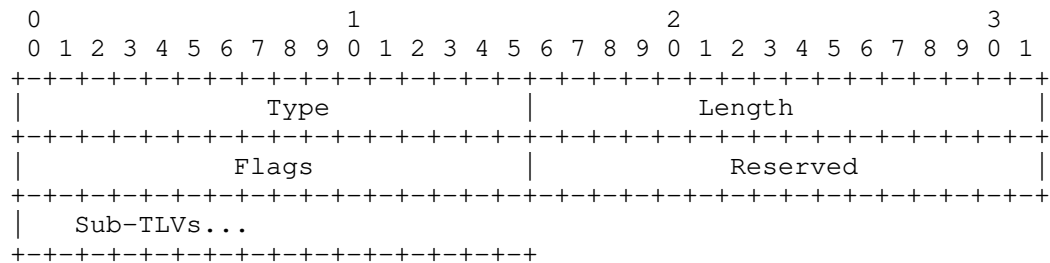
The SRv6 Capabilities TLV is designed for an OSPFv3 router to advertise its SRv6 support along with its related capabilities for SRv6 functionality. This is a new optional top level TLV of OSPFv3 Router Information LSA [RFC7770] which MUST be advertised by a SRv6 enabled router.

This TLV SHOULD be advertised only once in the OSPFv3 Router Information LSA. When multiple SRv6 Capabilities TLVs are received from a given router, the receiver MUST use the first occurrence of the TLV in the OSPFv3 Router Information Opaque LSA. If the SRv6 Capabilities TLV appears in multiple OSPFv3 Router Information Opaque

LSAs that have different flooding scopes, the TLV in the OSPFv3 Router Information Opaque LSA with the area-scoped flooding scope MUST be used. If the SRv6 Capabilities TLV appears in multiple OSPFv3 Router Information Opaque LSAs that have the same flooding scope, the TLV in the OSPFv3 Router Information Opaque LSA with the numerically smallest Instance ID MUST be used and subsequent instances of the TLV MUST be ignored.

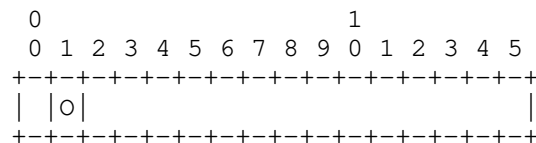
The OSPFv3 Router Information Opaque LSA can be advertised at any of the defined opaque flooding scopes (link, area, or Autonomous System (AS)). For the purpose of SRv6 Capabilities TLV advertisement, area-scoped flooding is REQUIRED.

The format of OSPFv3-SRv6-Capabilities TLV is shown below



Where:

- o Type: 16 bit field. TBD
- o Length: 16 bit field. Length of Capability TLV + length of Sub-TLVs
- o Reserved : 16 bit field. SHOULD be set to 0 and MUST be ignored by receiver.
- o Flags: 16 bit field. The following flags are defined:



where:

- * O-flag: If set, then router is capable of supporting SRH O-bit, as specified in [I-D.ali-spring-srv6-oam].

The SRv6 Capabilities TLV may contain optional sub-TLVs. No sub-TLVs are currently defined.

3. Advertisement of Supported Algorithms

SRv6 enabled OSPFv3 router advertises its algorithm support using the SR Algorithm TLV defined in [I-D.ietf-ospf-segment-routing-extensions] as described in [I-D.ietf-ospf-ospfv3-segment-routing-extensions].

4. Advertisement of SRH Operation Limits

A SRv6 enabled router may have different capabilities and limits when it comes to SRH processing and this needs to be advertised to other routers in the SRv6 domain.

[RFC8476] defines the means to advertise node/link specific values for Maximum SID Depths (MSD) of various types. Node MSDs are advertised using the Node MSD TLV in the OSPFv3 Router Information LSA [RFC7770] while Link MSDs are advertised using the Link MSD sub-TLV of the E-Router-LSA TLV [RFC8362]. The format of the MSD types for OSPFv3 is defined in [RFC8476].

The MSD types for SRv6 that are defined in [I-D.ietf-lsr-isis-srv6-extensions] for IS-IS are also used by OSPFv3. These MSD Types are allocated under the IGP MSD Types registry maintained by IANA that are shared by IS-IS and OSPF.

5. Advertisement of SRv6 Locator and End SIDs

An SRv6 Segment Identifier (SID) is 128 bits and represented as LOC:FUNCT as described in [I-D.ietf-spring-srv6-network-programming].

A node is provisioned with algorithm specific locators for each algorithm supported by that node. Each locator is a covering prefix for all SIDs provisioned on that node which have the matching algorithm.

Locators MUST be advertised in the SRv6 Locator LSA (see Section 6). Forwarding entries for the locators advertised in the SRv6 Locator LSA MUST be installed in the forwarding plane of receiving SRv6 capable routers when the associated algorithm is supported by the receiving node. Locators can be of different route types similar to existing OSPF LSA route types - Intra-Area, Inter-Area, External and NSSA. The computation of locator reachability and their advertisement are similar to how normal OSPF prefix reachability LSAs are processed as part of the SPF computation.

Locators are routable and MAY also be advertised via Prefix LSAs of different types - Inter-Area Prefix LSA, AS-External LSA, NSSA LSA or Intra-Area Prefix LSA (or their equivalent extended LSAs [RFC8362]). Locators associated with Flexible Algorithms SHOULD NOT be advertised via Prefix LSAs. Locators associated with algorithm 0 (for all supported topologies) SHOULD be advertised in Prefix LSAs 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 LSA and an SRv6 Locator LSA, the Prefix LSA 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 of E-Router-Link TLV.

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 algorithm will be forwarded correctly, while SRv6 SIDs associated with an unsupported algorithm will be dropped. NOTE: The drop behavior depends on the absence of a default/summary route covering a given locator.

6. SRv6 Locator LSA

The SRv6 Locator LSA has a function code of TBD while the S1/S2 bits are dependent on the desired flooding scope for the LSA. The flooding scope of the SRv6 Locator LSA depends on the scope of the advertised SRv6 Locator and is under the control of the advertising router. The U bit will be set indicating that the LSA should be flooded even if it is not understood.

Multiple SRv6 Locator LSAs can be advertised by an OSPFv3 router and they are distinguished by their Link State IDs (which are chosen arbitrarily by the originating router).

The format of SRv6 Locator LSA is shown below:

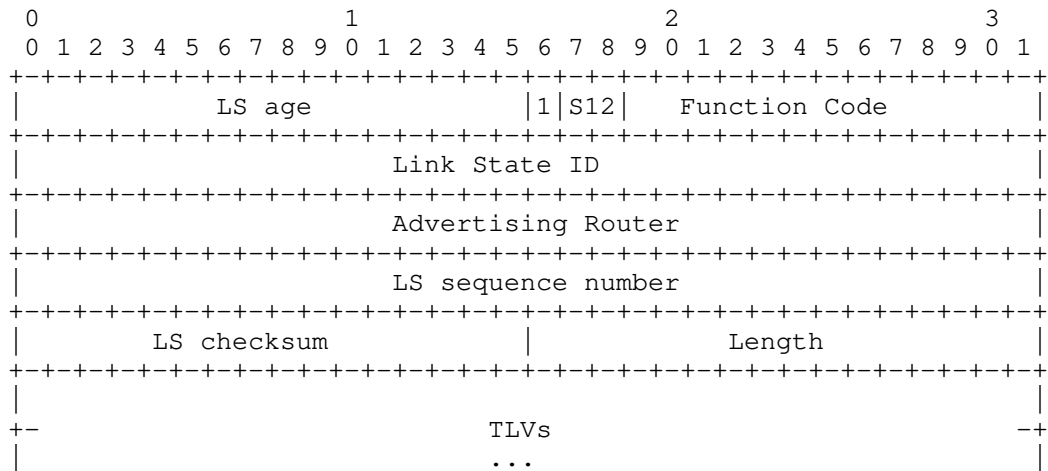


Figure 1: SRv6 Locator LSA

The format of the TLVs within the body of the SRv6 Locator LSA is the same as the format used by [RFC3630]. The variable TLV section consists of one or more nested TLV tuples. Nested TLVs are also referred to as sub-TLVs. The format of each TLV is:

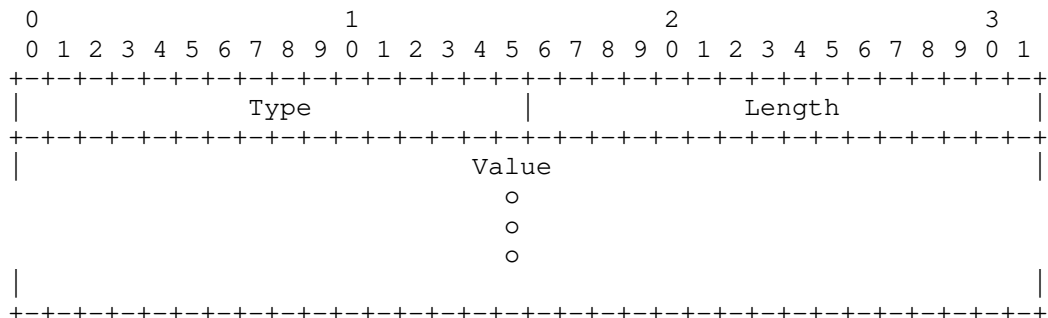


Figure 2: SRv6 Locator LSA TLV Format

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). Nested TLVs are also 32-bit aligned. For example, a 1-byte value would have the Length field set to 1, and 3 octets of padding would be added to the end of the value portion of the TLV. The padding is composed of zeros.

6.1. SRv6 Locator TLV

The SRv6 Locator TLV is a top-level TLV of the SRv6 Locator LSA that is used to advertise a SRv6 Locator, its attributes and SIDs associated with it. Multiple SRv6 Locator TLVs MAY be advertised in each SRv6 Locator LSA. However, since the S12 bits define the flooding scope, the LSA flooding scope MUST satisfy the application-specific requirements for all the locators included in a single SRv6 Locator LSA.

When multiple SRv6 Locator TLVs are received from a given router in a SRv6 Locator LSA for the same Locator, the receiver MUST use the first occurrence of the TLV in the LSA. If the SRv6 Locator TLV for the same Locator appears in multiple SRv6 Locator LSAs that have different flooding scopes, the TLV in the SRv6 Locator LSA with the area-scoped flooding scope MUST be used. If the SRv6 Locator TLV for the same Locator appears in multiple SRv6 Locator LSAs that have the same flooding scope, the TLV in the SRv6 Locator LSA with the numerically smallest Link-State ID MUST be used and subsequent instances of the TLV MUST be ignored.

The format of SRv6 Locator TLV is shown below:

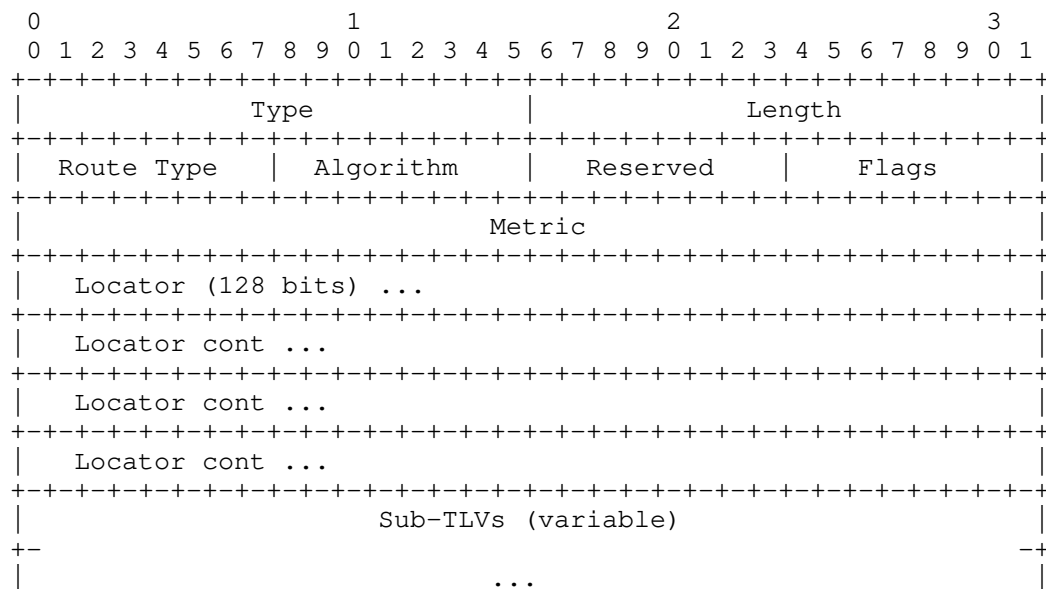


Figure 3: SRv6 Locator TLV

Where:

Type: 16 bit field. The value is 1 for this type.

Length: 16 bit field. The total length of the value portion of the TLV including sub-TLVs.

Route Type : 8 bit field. The type of the locator route. Supported types are the ones listed below and other other types MUST be ignored by the receiver.

- 1 - Intra-Area
- 2 - Inter-Area
- 3 - AS External
- 4 - NSSA External

Figure 4

Algorithm: 8 bit field. Associated algorithm. Algorithm values are defined in the IGP Algorithm Type registry.

Reserved: 8 bit field. SHOULD be set to 0 by sender and MUST be ignored by receiver.

Flags: 8 bit field. The following flags are defined

```

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

```

Figure 5

- * N flag : When the locator uniquely identifies a node in the network (i.e. it is provisioned on one and only one node), the N bit MUST be set. Otherwise, this bit MUST be clear.
- * A bit : When the Locator is configured as anycast, the A bit SHOULD be set. Otherwise, this bit MUST be clear.
- * Other flags are not defined and SHOULD be set to 0 and MUST be ignored on receipt.

Metric : 32 bit field. The metric value associated with the locator.

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

Sub-TLVs : Used to advertise sub-TLVs that provide additional attributes for the given SRv6 Locator and SRv6 SIDs associated with it.

7. Advertisement of SRv6 End SIDs

SRv6 End SID sub-TLV is a new sub-TLV of SRv6 Locator TLV in the SRv6 Locator LSA (defined in Section 6). It is used to advertise the SRv6 SIDs belonging to the node along with their associated functions. SIDs associated with adjacencies are advertised as described in Section 8. Every SRv6 enabled OSPFv3 router SHOULD advertise at least one SRv6 SID associated with an END behavior for its node as specified in [I-D.ietf-spring-srv6-network-programming].

SRv6 End SIDs inherit the algorithm from the parent locator. 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.

The router MAY advertise multiple instances of the SRv6 End SID sub-TLV within the SRv6 Locator TLV - one for each of the SRv6 SIDs to be advertised. When multiple SRv6 End SID sub-TLVs are received in the SRv6 Locator TLV from a given router for the same SRv6 SID value, the receiver MUST use the first occurrence of the sub-TLV in the SRv6 Locator TLV.

The format of SRv6 End SID sub-TLV is shown below

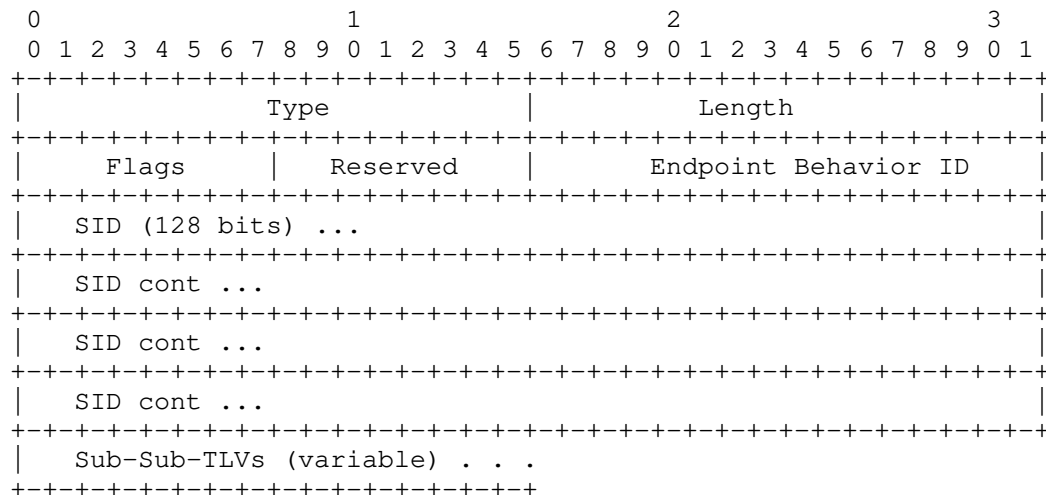


Figure 6: SRv6 End SID sub-TLV

Where:

Type: 16 bit field. Value is 1 for this type.

Length: 16 bit field. The total length of the value portion of the sub-TLV including sub-sub-TLVs.

Reserved : 8 bit field. Should be set to 0 and MUST be ignored on receipt.

Flags: 8 bit field which define the flags associated with the SID. No flags are currently defined and SHOULD be set to 0 and MUST be ignored on receipt.

Endpoint Behavior ID: 16 bit field. The endpoint behavior code point for this SRv6 SID as defined in [I-D.ietf-spring-srv6-network-programming].

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

Sub-Sub-TLVs : Used to advertise sub-sub-TLVs that provide additional attributes for the given SRv6 SID.

8. Advertisement of SRv6 SIDs Associated with Adjacencies

The SRv6 endpoint behaviors are defined in [I-D.ietf-spring-srv6-network-programming] include certain behaviors which are specific to links or adjacencies. The most basic of this which is critical for link state routing protocols like OSPFv3 is the End.X behavior that is an instruction to forward to a specific neighbor on a specific link. These SRv6 SIDs along with others that are defined in [I-D.ietf-spring-srv6-network-programming] which are specific to links or adjacencies need to be advertised by OSPFv3 so that this information is available with all routers in the area to influence the packet path via these SRv6 SIDs over the specific adjacencies.

The advertising of SRv6 SIDs and their behaviors that are specific to a particular neighbor are done via two different optional sub-TLVs of the E-Router-Link TLV defined in [RFC8362] as follows:

- o SRv6 End.X SID Sub-TLV: for OSPFv3 adjacency over point-to-point or point-to-multipoint links and the adjacency to the Designated Router (DR) over broadcast and non-broadcast-multi-access (NBMA) links.
- o SRv6 LAN End.X SID Sub-TLV: for OSPFv3 adjacency on broadcast and NBMA links to the Backup DR and DR-Other neighbors. This sub-TLV

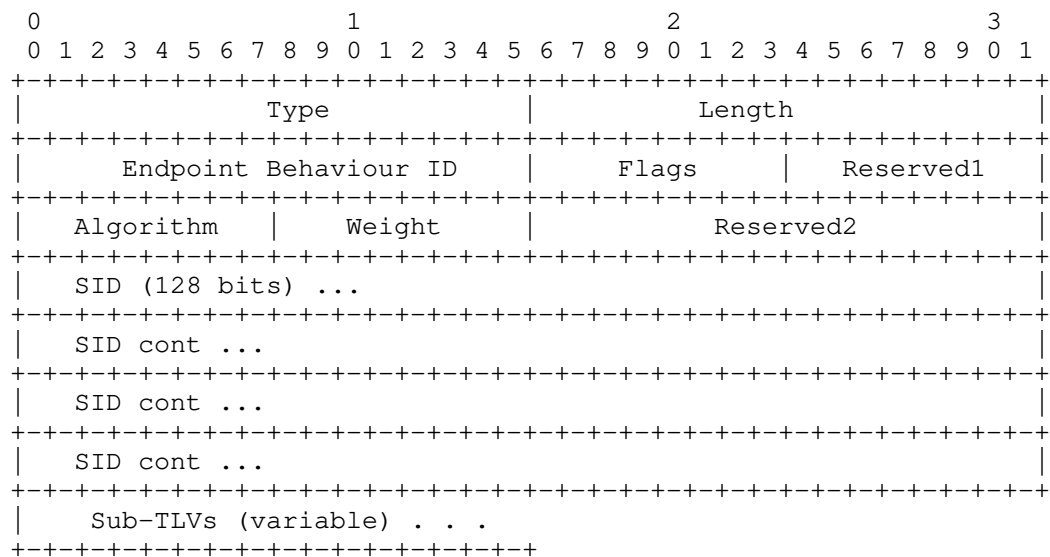
includes the OSPFv3 router-id of the neighbor and thus allows for multiple instances of this TLV for each neighbor to be explicitly advertised under the E-Router-Link TLV for the same link.

Every SRv6 enabled OSPFv3 router SHOULD instantiate at least one End.X function with a unique SRv6 SID corresponding to each of its neighbor. A router MAY instantiate more than one SRv6 SID for the End.X function for a single neighbor. The same SRv6 SID MAY be advertised for the End.X function corresponding to more than one neighbor. Thus multiple instances of the SRv6 End.X SID and SRv6 LAN End.X SID sub-TLVs MAY be advertised within the E-Router-Link TLV for a single link.

All End.X SIDs MUST be a subnet of a Locator with matching 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.

8.1. SRv6 End.X SID Sub-TLV

The format of the SRv6 End.X SID sub-TLV is shown below



Where:

Type is TBD

Length: 16 bit field. The total length of the value portion of the TLV.

Endpoint Behaviour ID: 16 bit field. The code point for the endpoint behavior for this SRv6 SID as defined in [I-D.ietf-spring-srv6-network-programming].

Flags: 8 bit field with the following definition:

```

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

```

- * B-Flag: Backup Flag. If set, the SID refers to a path that is eligible for protection.
- * 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: If set, the SID is persistently allocated, i.e., the SID value remains consistent across router restart and session/interface flap.
- * Rsvd bits: Reserved for future use and MUST be zero when originated and ignored when received.

Reserved1 : 8 bit field. Should be set to 0 and MUST be ignored on receipt.

Algorithm : 8 bit field. Associated algorithm. Algorithm values are defined in the IGP Algorithm Type registry.

Weight: 8 bit field whose value represents the weight of the End.X SID for the purpose of load balancing. The use of the weight is defined in [RFC8402].

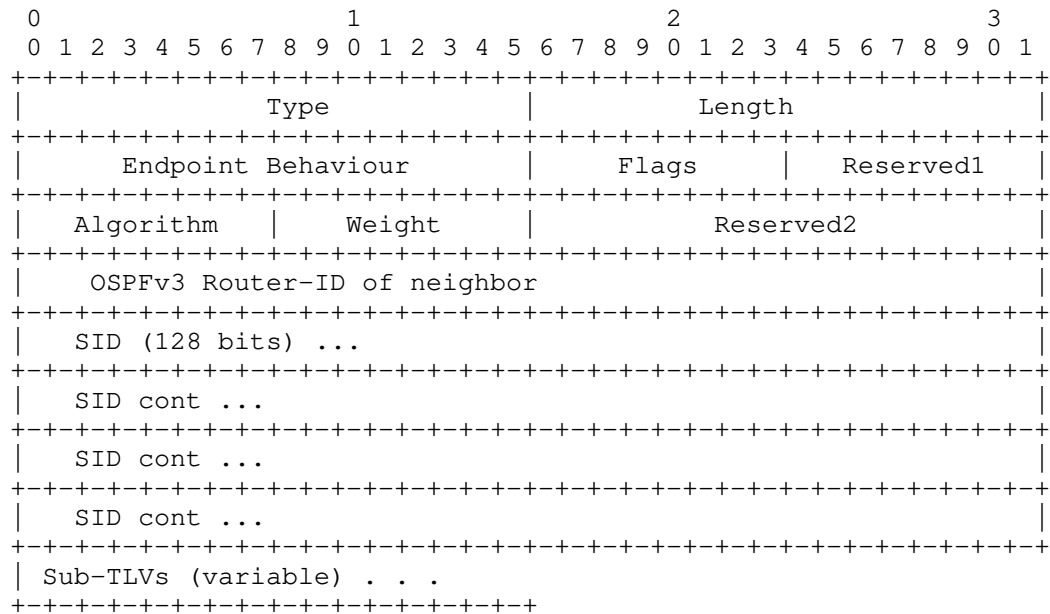
Reserved2 : 16 bit field. Should be set to 0 and MUST be ignored on receipt.

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

Sub-TLVs : Used to advertise sub-TLVs that provide additional attributes for the given SRv6 End.X SID.

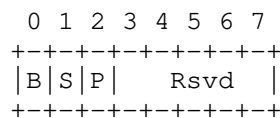
8.2. SRv6 LAN End.X SID Sub-TLV

The format of the SRv6 LAN End.X SID sub-TLV is as shown below



Where

- o Type: TBD
- o Length: 16 bit value. Variable
- o Endpoint Behaviour: 16 bit field. The code point for the endpoint behavior for this SRv6 SID as defined in [I-D.ietf-spring-srv6-network-programming].
- o SID Flags: 8 bit field which define the flags associated with the SID. No flags are currently defined and SHOULD be set to 0 and MUST be ignored on receipt.
- o Flags: 8 bit field with the following definition:



- * B-Flag: Backup Flag. If set, the SID refers to a path that is eligible for protection.
 - * 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: If set, the SID is persistently allocated, i.e., the SID value remains consistent across router restart and session/interface flap.
 - * Rsvd bits: Reserved for future use and MUST be zero when originated and ignored when received.
- o Reserved1 : 8 bit field. Should be set to 0 and MUST be ignored on receipt.
 - o Algorithm : 8 bit field. Associated algorithm. Algorithm values are defined in the IGP Algorithm Type registry.
 - o Weight: 8 bit field whose value represents the weight of the End.X SID for the purpose of load balancing. The use of the weight is defined in [RFC8402].
 - o Reserved2 : 16 bit field. Should be set to 0 and MUST be ignored on receipt.
 - o Neighbor ID : 4 octets of OSPFv3 Router-id of the neighbor
 - o SID: 16 octets. This field encodes the advertised SRv6 SID.
 - o Sub-TLVs : Used to advertise sub-TLVs that provide additional attributes for the given SRv6 SID.

9. SRv6 SID Structure sub-TLV

SRv6 SID Structure sub-TLV is used to advertise the length of each individual part of the SRv6 SID as defined in [I-D.ietf-spring-srv6-network-programming]. It is used as an optional sub-sub-TLV of the following:

- o SRv6 End SID sub-TLV (refer Section 7)
- o SRv6 End.X SID sub-TLV (refer Section 8.1)
- o SRv6 LAN End.X SID sub-TLV (refer Section 8.2)

The sub-TLV has the following format:

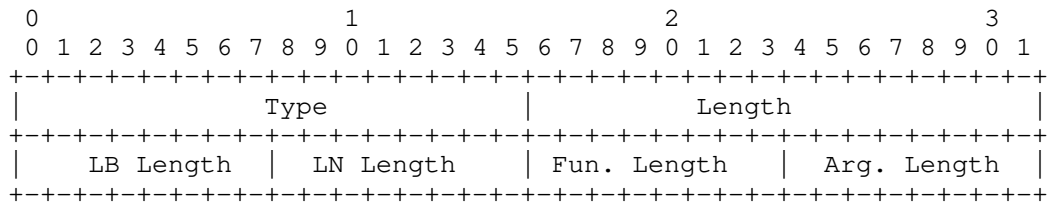


Figure 7: SRv6 SID Structure sub-TLV

Where:

Type: 2 octet field with value TBD, see Section 11.

Length: 2 octet field with the value 4.

LB Length: 1 octet field. SRv6 SID Locator Block length in bits.

LN Length: 1 octet field. SRv6 SID Locator Node length in bits.

Function Length: 1 octet field. SRv6 SID Function length in bits.

Argument Length: 1 octet field. SRv6 SID Argument length in bits.

10. Security Considerations

Existing security extensions as described in [RFC5340] and [RFC8362] apply to these SRv6 extensions. While OSPFv3 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.

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.

11. IANA Considerations

This document specifies updates to multiple OSPF and OSPFv3 related IANA registries as follows.

11.1. OSPF Router Information TLVs

This document proposes the following new code point in the "OSPF Router Information (RI) TLVs" registry under the "OSPF Parameters" registry for the new TLVs:

- Type TBD (suggested 17): SRv6-Capabilities TLV: Refer to Section 2.

11.2. OSPFv3 LSA Function Codes

This document proposes the following new code point in the "OSPFv3 LSA Function Codes" registry under the "OSPFv3 Parameters" registry for the new SRv6 Locator LSA:

- o Type TBD (suggested 42): SRv6 Locator LSA: Refer to Section 6.

11.3. OSPFv3 Extended-LSA sub-TLVs

This document proposes the following new code points in the "OSPFv3 Extended-LSA Sub-TLVs" registry under the "OSPFv3 Parameters" registry for the new sub-TLVs:

- o Type TBD (suggested 10): SRv6 SID Structure Sub-TLV : Refer to Section 9.
- o Type TBD (suggested 11): SRv6 End.X SID Sub-TLV : Refer to Section 8.1.
- o Type TBD (suggested 12): SRv6 LAN End.X SID Sub-TLV : Refer to Section 8.2.

11.4. OSPFv3 Locator LSA TLVs

This document proposes setting up of a new "OSPFv3 Locator LSA TLVs" registry that defines top-level TLVs for the OSPFv3 SRv6 Locator LSA to be added under the "OSPFv3 Parameters" registry. The initial code-points assignment is as below:

- o Type 0: Reserved.
- o Type 1: SRv6 Locator TLV : Refer to Section 6.1.

Types in the range 2-32767 are allocated via IETF Review or IESG Approval [RFC8126].

Types in the range 32768-33023 are Reserved for Experimental Use; these will not be registered with IANA and MUST NOT be mentioned by RFCs.

Types in the range 33024-45055 are to be assigned on a First Come First Served (FCFS) basis.

Types in the range 45056-65535 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 cover the range being assigned.

11.5. OSPFv3 Locator LSA sub-TLVs

This document proposes setting up of a new "OSPFv3 Locator LSA Sub-TLVs" registry that defines sub-TLVs at any level of nesting for the SRv6 Locator TLVs to be added under the "OSPFv3 Parameters" registry. The initial code-points assignment is as below:

- o Type 0: Reserved.
- o Type 1: SRv6 End SID sub-TLV : Refer to Section 7.
- o Type 10: SRv6 SID Structure Sub-TLV : Refer to Section 9.

Types in the range 2-9 and 11-32767 are allocated via IETF Review or IESG Approval [RFC8126].

Types in the range 32768-33023 are Reserved for Experimental Use; these will not be registered with IANA and MUST NOT be mentioned by RFCs.

Types in the range 33024-45055 are to be assigned on a First Come First Served (FCFS) basis.

Types in the range 45056-65535 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 cover the range being assigned.

12. Acknowledgements

TBD

13. References

13.1. 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.ietf-6man-segment-routing-header]
Filsfils, C., Dukes, D., Previdi, S., Leddy, J.,
Matsushima, S., and d. daniel.voyer@bell.ca, "IPv6 Segment
Routing Header (SRH)", draft-ietf-6man-segment-routing-
header-21 (work in progress), June 2019.
- [I-D.ietf-lsr-isis-srv6-extensions]
Psenak, P., Filsfils, C., Bashandy, A., Decraene, B., and
Z. Hu, "IS-IS Extension to Support Segment Routing over
IPv6 Dataplane", draft-ietf-lsr-isis-srv6-extensions-02
(work in progress), July 2019.
- [I-D.ietf-ospf-ospfv3-segment-routing-extensions]
Psenak, P. and S. Previdi, "OSPFv3 Extensions for Segment
Routing", draft-ietf-ospf-ospfv3-segment-routing-
extensions-23 (work in progress), January 2019.
- [I-D.ietf-ospf-segment-routing-extensions]
Psenak, P., Previdi, S., Filsfils, C., Gredler, H.,
Shakir, R., Henderickx, W., and J. Tantsura, "OSPF
Extensions for Segment Routing", draft-ietf-ospf-segment-
routing-extensions-27 (work in progress), December 2018.
- [I-D.ietf-spring-srv6-network-programming]
Filsfils, C., Camarillo, P., Leddy, J.,
daniel.voyer@bell.ca, d., Matsushima, S., and Z. Li, "SRv6
Network Programming", draft-ietf-spring-srv6-network-
programming-01 (work in progress), July 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>>.

- [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>>.
- [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>>.
- [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>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [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>>.
- [RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.
- [RFC8476] Tantsura, J., Chunduri, U., Aldrin, S., and P. Psenak, "Signaling Maximum SID Depth (MSD) Using OSPF", RFC 8476, DOI 10.17487/RFC8476, December 2018, <<https://www.rfc-editor.org/info/rfc8476>>.

13.2. Informative References

- [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>>.

Authors' Addresses

Zhenbin Li
Huawei Technologies
Email: lizhenbin@huawei.com

Zhibo Hu
Huawei Technologies
Email: huzhibo@huawei.com

Dean Cheng
Huawei Technologies
Email: dean.cheng@huawei.com

Ketan Talaulikar
Cisco Systems
India
Email: ketant@cisco.com

Peter Psenak
Cisco Systems
Slovakia
Email: ppsenak@cisco.com