Internet
Internet-Draft
Intended status: Standards Track
Expires: January 13, 2021

D. Yeung
Arrcus
Y. Qu
Futurewei
J. Zhang
Juniper Networks
I. Chen
The MITRE Corporation
A. Lindem
Cisco Systems
July 12, 2020

YANG Data Model for OSPF SR (Segment Routing) Protocol
draft-ietf-ospf-sr-yang-12

Abstract

This document defines a YANG data model that can be used to configure and manage OSPF Segment Routing. The model is based on YANG 1.1 as defined in RFC 7950 and conforms to the Network Management Datastore Architecture (NDMA) as described in RFC 8342.

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, 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. Overview 2
  1.1. Requirements Language 2
2. Tree Diagrams 3
3. OSPF Segment Routing 3
4. OSPF Segment Routing YANG Module 7
5. Security Considerations 20
6. Acknowledgements 20
7. References 21
  7.1. Normative References 21
  7.2. Informative References 22
Appendix A. Contributors' Addreses 24
Authors' Addresses 24

1. Overview

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

This document defines a YANG data model that can be used to configure and manage OSPF Segment Routing and it is an augmentation to the OSPF 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", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. Tree Diagrams

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

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 [I-D.ietf-spring-sr-yang], 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
    |     +--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/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 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
            /ospf:extended-prefix-tlv:
      +--ro perfix-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 perfix-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 perfix-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:
      +--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 perfix-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
      +--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:
      +--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 perfix-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
      +--ro sr-algorithm-tlv
      |  +--ro sr-algorithm*   uint8
      +--ro sid-range-tlvs
      |  +--ro sid-range-tlv*
      |     +--ro range-size?    uint24
      |     +--ro sid-sub-tlv
      |        +--ro sid?   uint32
      +--ro local-block-tlvs
      |  +--ro local-block-tlv*
      |     +--ro range-size?    uint24
      |     +--ro sid-sub-tlv
      |        +--ro sid?   uint32
      +--ro srms-preference-tlv
         +--ro preference?   uint8
    augment /rt:routing/rt:control-plane-protocols
            /rt:control-plane-protocol/ospf:ospf/ospf:database
            /ospf:as-scope-lsa-type/ospf:as-scope-lsas/ospf:as-scope-lsa
            /ospf:version/ospf:ospfv2/ospf:ospfv2/ospf:body/ospf:opaque:
      +--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 perfix-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
      +--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
```

4. OSPF Segment Routing YANG Module

```
<CODE BEGINS> file "ietf-ospf-sr@2020-02-05.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";
    }
    import ietf-segment-routing-mpls {
      prefix "sr-mpls";
    }
    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@arrcus.com>
        Author:   Derek Yeung
                  <mailto:derek@arrcus.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 8242.

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

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

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

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

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

    reference "RFC XXXX";

    revision 2020-07-12 {
      description
        "Initial revision.";
      reference
        "RFC XXXX: 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 perfix-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 {    //type=2?
          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, vlaue 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-controlplane;
      container protocol-srgb {
        if-feature sr-mpls:protocol-srgb;
        uses sr-cmn:srgb;
        description
          "Per-protocol SRGB.";
      }
    }

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

```
      when "../../../../../../../../../../../../"
         + "rt:type = 'ospf:ospfv2'" {
        description
          "This augmentation is only valid for OSPFv2.";
      }

      description
        "SR specific TLVs for OSPFv2 type 9 opaque LSA.";

      uses extended-prefix-range-tlvs;
      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" {
      when "../../../../../../../../../../../"
         + "rt:type = 'ospf:ospfv2'" {
        description
          "This augmentation is only valid for OSPFv2.";
      }

      description
        "SR specific TLVs for OSPFv2 type 10 opaque LSA.";

      uses extended-prefix-range-tlvs;
      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" {
      when "../../../../../../../../../"
         + "rt:type = 'ospf:ospfv2'" {
        description
```

```
          "This augmentation is only valid for OSPFv2.";
      }
      description
        "SR specific TLVs for OSPFv2 type 11 opaque LSA.";

      uses extended-prefix-range-tlvs;
      uses sr-algorithm-tlv;
      uses sid-range-tlvs;
      uses local-block-tlvs;
      uses srms-preference-tlv;
    }
  }
  <CODE ENDS>
```

5. Security Considerations

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

6. 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.

7. References

7.1. Normative 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-29 (work in progress), October 2019.

[I-D.ietf-spring-sr-yang]
Litkowski, S., Qu, Y., Lindem, A., Sarkar, P., and J. Tantsura, "YANG Data Model for Segment Routing", draft-ietf-spring-sr-yang-17 (work in progress), July 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>.

[RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <https://www.rfc-editor.org/info/rfc2328>.

[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>.

[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>.

[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>.

[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>.

[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>.

[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>.

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
USA

EMail: yingzhen.qu@futurewei.com

Jeffrey Zhang
Juniper Networks
10 Technology Park Drive
Westford, MA 01886
USA

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