Workgroup: Network Working Group
Internet-Draft:
draft-ietf-lsr-ospf-srv6-yang-00
Published: 18 August 2021
Intended Status: Standards Track
Expires: 19 February 2022
Authors: Z. Hu
Huawei Technologies
K. Raza
Cisco Systems, Inc.
A. Lindem
Cisco Systems
X. Geng
Huawei Technologies
Y. Qu
Futurewei Technologies

YANG Data Model for OSPF SRv6

Abstract

This document defines a YANG data model that can be used to configure and manage OSPFv3 SRv6 as defined in I-D.ietf-lsr-ospfv3-srv6-extensions.

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 19 February 2022.

Copyright Notice

Table of Contents

1. Introduction

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 OSPFv3 SRv6 and it is an augmentation to the OSPF YANG data model.

2. Terminology and Notation

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 following terms are defined in [RFC8342]:

- client
- server
- configuration
- system state
- operational state
- intended configuration

The following terms are defined in [RFC7950]:

- action
- augment
- container
- container with presence
- data model
- data node
- feature
- leaf
- list
- mandatory node
- module
- schema tree
- RPC (Remote Procedure Call) operation

2.1. Tree Diagrams

Tree diagrams used in this document follow the notation defined in [RFC8340].

2.2. Prefixes in Data Node Names

In this document, names of data nodes, actions, and other data model objects are often used without a prefix, as long as it is clear from the context in which YANG module each name is defined. Otherwise, names are prefixed using the standard prefix associated with the corresponding YANG module, as shown in Table 1.

Prefix	YANG Module	Reference
ospfv3-e-lsa	ietf-ospfv3-extended-lsa	I-D.ietf-lsr-ospfv3-extended-lsa-yang
ospf	ietf-ospf	I-D.ietf-ospf-yang
srv6	ietf-srv6-base	I-D.ietf-spring-srv6-yang
sr	ietf-segment-routing	[RFC9020]
rt	ietf-routing	[RFC8349]
yang	ietf-yang-types	[RFC6991]
inet	ietf-inet-types	[RFC6991]

Figure 1: Table 1: Prefixes and Corresponding YANG Modules

3. OSPFv3 SRv6 Configuration

This document defines a model for OSPFv3 SRv6 feature. It is an augmentation of the OSPF base model.

The OSPFv3 SRv6 YANG module requires support of OSPF base model [I-D.ietf-ospf-yang] which defines basic OSPF configuration and state and support of OSPFv3 Extended LSAs model [I-D.ietf-lsr-ospfv3-extended-lsa-yang].

3.1. SRv6 Activation

Activation of OSPFv3 SRv6 is done by setting the "enable" leaf to true. This triggers advertisement of SRv6 extensions based on the

configuration parameters that have been setup using the base SRv6 module [I-D.ietf-spring-srv6-yang].

3.2. Locator Setting

The SRv6 base module [I-D.ietf-spring-srv6-yang] defines locators. When OSPFv6 SRv6 is enabled, the specified locators are used unless it is enabled to use the default locator. The default locator can be set by using two leafs, i.e.,"default-locator" leaf, "locator-name" leaf.

3.3. IP Fast Reroute

Additional data object (in the OSPFv3 SRv6 model) for fast reroute [RFC5286] is introduced by augmenting the fast-reroute container of the OSPF module. It brings the ability to activate ipv6 TI-LFA (topology independent LFA) [I-D.ietf-rtgwg-segment-routing-ti-lfa].

3.4. Micro-loop Avoidance

OSPFv3 SRv6 model augments OSPF module with the micro-loop-avoidance container, this container including the leaf "srv6-enable" brings the ability to activate SRv6 avoid-microloop.

4. YANG Module and Tree

4.1. OSPFv3 SRv6 Model Tree

The figure below describes the overall structure of the ospfv3-srv6 YANG module:

```
module: ietf-ospfv3-srv6
  augment /rt:routing/rt:control-plane-protocols
          /rt:control-plane-protocol/ospf:ospf:
     +--rw srv6
     |  +--rw enable?                boolean
     |  +--rw default-locator?       boolean
     |  +--rw locator-name*    -> /rt:routing/srv6:srv6
     |                            /srv6:locators/srv6:locator/srv6:name
     |  +--rw persistent-end-x-sid?   boolean
     +--rw micro-loop-avoidance
        +--rw srv6-enable?            boolean
        +--rw srv6-rib-update-delay?  uint16
  augment /rt:routing/rt:control-plane-protocols
          /rt:control-plane-protocol/ospf:ospf
          /ospf:fast-reroute:
     +--rw srv6-ti-lfa {srv6-ti-lfa}?
        +--rw enable?   boolean
  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 srv6-locator
        +--ro srv6-locactor-tlv
           +--ro route-type?       identityref
           +--ro algorithm?        uint8
           +--ro locator-length?   uint8
           +--ro flags?            bits
           +--ro metric?           uint32
           +--ro locator*          inet:ipv6-address-no-zone
           +--ro srv6-end-sid
              +--ro flags?                uint8
              +--ro endpoint-func
              |  +--ro fuc-flags?                uint8
              |  +--ro endpoint-func?            identityref
              |  +--ro undefined-endpoint-func?   uint16
              +--ro sid?                  srv6-sid-value
              +--ro srv6-sid-structure
                 +--ro lb-length?    uint8
                 +--ro ln-length?    uint8
                 +--ro fun-length?   uint8
                 +--ro arg-length?   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 srv6-capability
   |  +--ro flags?   bits
   +--ro srv6-msd
      +--ro max-segments-left?   uint8
      +--ro max-end-pop?         uint8
      +--ro max-h-encaps?        uint8
      +--ro max-end-d?           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:ospfv3-e-lsa:e-router
        /ospfv3-e-lsa:e-router-tlvs
        /ospfv3-e-lsa:link-tlv:
   +--ro srv6-endx-sid
      +--ro endpoint-func
      |  +--ro fuc-flags?                uint8
      |  +--ro endpoint-func?            identityref
      |  +--ro undefined-endpoint-func?  uint16
      +--ro flags?               bits
      +--ro algorithm?           uint8
      +--ro weight?              uint8
      +--ro sid*                 srv6-sid-value
      +--ro neighbor-router-id?  yang:dotted-quad
      +--ro srv6-sid-structure
         +--ro lb-length?   uint8
         +--ro ln-length?   uint8
         +--ro fun-length?  uint8
         +--ro arg-length?  uint8
```

4.2. OSPFv3 SRv6 YANG Module

```
<CODE BEGINS> file "ietf-ospfv3-srv6@2020-09-29.yang"

module ietf-ospfv3-srv6 {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:"
          + "yang:ietf-ospfv3-srv6";
  prefix ospfv3-srv6;

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

  import ietf-routing {
    prefix "rt";
  }

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

  import ietf-ospf {
    prefix "ospf";
  }

  import ietf-srv6-base {
    prefix "srv6";
  }

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

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

  organization
   "IETF LSR Working Group";

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

     Author:    Zhibo Hu
                <mailto:huzhibo@huawei.com>
     Author:    Xuesong Geng
                <mailto:gengxuesong@huawei.com>
     Author:    Kamran Raza
                <mailto:skraza@cisco.com>
     Author:    Yingzhen Qu
```

```
              <mailto:yingzhen.qu@futurewei.com>
   Author:    Acee Lindem
              <mailto:acee@cisco.com>
  ";

description
  "The YANG module defines a generic configuration model for
   Segment IPV6 routing OSPFv3 extensions common across all
   of the vendor implementations.

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

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

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

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

revision 2020-09-29 {
  description
    "Initial revision.";
  reference "draft-ietf-lsr-ospfv3-srv6-extensions-01";
}

/* Identities */
identity SRV6_END_FUNC_TYPE {
  description
    "Base identity type for srv6 endpoint function code points.";
}

identity SRV6_END_FUNC_NO_PSP_USP {
  base "SRV6_END_FUNC_TYPE";
  description
    "End (no PSP, no USP).";
}
```

```
identity SRV6_END_FUNC_PSP {
  base "SRV6_END_FUNC_TYPE";
  description
    "End with PSP.";
}

identity SRV6_END_FUNC_USP {
  base "SRV6_END_FUNC_TYPE";
  description
    "END with USP.";
}

identity SRV6_END_FUNC_PSP_USP {
  base "SRV6_END_FUNC_TYPE";
  description
    "END with PSP & USP.";
}

identity SRV6_END_T_FUNC_NO_PSP_USP {
  base "SRV6_END_FUNC_TYPE";
  description
    "End.T (no PSP, no USP).";
}

identity SRV6_END_T_FUNC_PSP {
  base "SRV6_END_FUNC_TYPE";
  description
    "End.T with PSP.";
}

identity SRV6_END_T_FUNC_USP {
  base "SRV6_END_FUNC_TYPE";
  description
    "End.T with USP.";
}

identity SRV6_END_T_FUNC_PSP_USP {
  base "SRV6_END_FUNC_TYPE";
  description
    "End.T with PSP & USP.";
}

identity SRV6_END_X_FUNC_NO_PSP_USP {
  base "SRV6_END_FUNC_TYPE";
  description
    "End.x (no PSP, no USP).";
}

identity SRV6_END_X_FUNC_PSP {
```

```
  base "SRV6_END_FUNC_TYPE";
  description
    "End.x with PSP.";
}

identity SRV6_END_X_FUNC_USP {
  base "SRV6_END_FUNC_TYPE";
  description
    "End.x with USP.";
}

identity SRV6_END_X_FUNC_PSP_USP {
  base "SRV6_END_FUNC_TYPE";
  description
    "End.x with PSP & USP.";
}

identity SRV6_END_FUNC_DX6 {
  base "SRV6_END_FUNC_TYPE";
  description
    "End.DX6 function.";
}

identity SRV6_END_FUNC_DT6 {
  base "SRV6_END_FUNC_TYPE";
  description
    "End.DT6 function.";
}

identity SRV6_END_FUNC_OTP {
  base "SRV6_END_FUNC_TYPE";
  description
    "END.OTP.";
}

identity s1-bit {
  base ospf:ospfv3-lsa-option;
  description
    "the S1/S2 bits are dependent on the desired
     flooding scope for the LSA.";
}

identity s2-bit {
  base ospf:ospfv3-lsa-option;
  description
    "the S1/S2 bits are dependent on the desired
     flooding scope for the LSA.";
}

identity srv6-locator-lsa {
```

```
  base ospf:ospfv3-lsa-type;
  description
    "SRv6 Locator LSA - Type TBD";
}

identity LOCATOR-ROUTE-TYPE {
  description
    "The type of the locator route.";
}

identity INTRA-AREA-LOCATOR {
  base "LOCATOR-ROUTE-TYPE";
  description
    "Intra-Area";
}

identity INTER-AREA-LOCATOR {
  base "LOCATOR-ROUTE-TYPE";
  description
    "Inter-Area";
}

identity AS-EXTERNAL-LOCATOR {
  base "LOCATOR-ROUTE-TYPE";
  description
    "AS External";
}

identity NSSA-EXTERNAL-LOCATOR {
  base "LOCATOR-ROUTE-TYPE";
  description
    "NSSA External";
}

/* typedef */
typedef srv6-sid-value {
  type inet:ipv6-address-no-zone;
  description
    "16 Octets encoded sid value.";
}

/* Features */
feature srv6-ti-lfa {
  description
    "Enhance SRv6 FRR with ti-lfa
     support";
}

/* Groupings */
grouping srv6-msds {
```

```
  description
    "Used to advertise node/link specific
     values for Maximum Sid Depths(MSD) of various types";
  container srv6-msd {
    description
    "Maximum SRv6 SID Depths.";
    leaf max-segments-left {
      type uint8;
      description
      "The Maximum Segments Left MSD Type specifies
       the maximum value of the 'SL' field in the SRH
       of a received packet before applying the Endpoint
       behavior associated with a SID.";
    }
    leaf max-end-pop {
      type uint8;
      description
      "The Maximum End Pop MSD Type specifies the maximum
       number of SIDs in the SRH to which the router can
       apply 'PSP' or 'USP' behavior, as defined in flavors.";
    }
    leaf max-h-encaps {
      type uint8;
      description
      "The Maximum H.Encaps MSD Type specifies the maximum number
       of SIDs that can be included as part of the 'H.Encaps'
       behavior";
    }
    leaf max-end-d {
      type uint8;
      description
        "The maximum number of SIDs in an SRH when performing
         decapsulation associated with 'End.Dx' functions
         (e.g., 'End.DX6' and 'End.DT6').";
    }
  }
}

grouping srv6-capabilities {
  description
    "SRV6 capability grouping.";
  container srv6-capability {
    description
      "SRv6 capability.";
    leaf flags {
      type bits {
        bit o-flag {
          position 1;
          description
```

```
            "If set, then router is capable of supporting SRH O-bit
            ";
        }
      }
      description
        "Flags.";
     reference
       "I-D.ali-spring-srv6-oam: OAM in SRv6";
    }
  }
}

grouping srv6-endpoint-func {
  description
    "This group defines srv6 endpoint function";
  container endpoint-func {
    description
      "Srv6 Endpoint function Descriptor.";
    leaf fuc-flags {
      type uint8;
      description
        "No function flags are currently being defined.";
    }
    leaf endpoint-func {
      type identityref {
        base SRV6_END_FUNC_TYPE;
      }
      description
        "The endpoint function.";
    }
    leaf undefined-endpoint-func {
      type uint16;
      description
        "Unknown endpoint func value.";
    }
  }
}

grouping srv6-end-sids {
  description
    "This group defines srv6 end sid";
  container srv6-end-sid {
    description
      "SRv6 Segment Identifier(SID) with Endpoint functions.";
    leaf flags {
      type uint8;
      description
        "NO flags are currently being defined.";
    }
```

```
    uses srv6-endpoint-func;
    leaf sid {
      type srv6-sid-value;
      description
        "SRV6 sid value.";
    }
    uses srv6-sid-structures;
  }
}

grouping srv6-sid-structures {
  description
    "This group defines SRv6 SID Structure sub-TLV.";
  container srv6-sid-structure {
    description
      "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]";
    leaf lb-length {
      type uint8;
      description
        "SRv6 SID Locator Block length in bits.";
    }

    leaf ln-length {
      type uint8;
      description
        "SRv6 SID Locator Node length in bits.";
    }

    leaf fun-length {
      type uint8;
      description
        "SRv6 SID Function length in bits.";
    }

    leaf arg-length {
      type uint8;
      description
        "SRv6 SID Argument length in bits.";
    }
  }
}

grouping srv6-endx-sids {
  description
    "This group defines SRv6 SIDs Associated with Adjacencies
    including SRv6 End.X SID Sub-TLV and SRv6 LAN End.X SID
```

```
  Sub-TLV.";
container srv6-endx-sid {
  description
    "SRv6 sids associated with an adjacency.";

  uses srv6-endpoint-func;

  leaf flags {
    type bits {
      bit b-flag {
        position 0;
        description
          "Backup Flag.  If set, the SID refers to a path that is
           eligible for protection";
      }
      bit s-flag {
        position 1;
        description
          "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).";
      }
      bit p-flag {
        position 2;
        description
          "Persistent Flag: If set, the SID is persistently
           allocated, i.e., the SID value remains consistent
           across router restart and session/interface flap.";
      }
    }
    description
      "Flags for end.x subtlv.";
  }

  leaf algorithm {
    type uint8;
    description
      "Associated algorithm.";
  }

  leaf weight {
    type uint8;
    description
      "8 bit field whose value represents the weight of the End.X
       SID for the purpose of load balancing";
  }

  leaf-list sid {
    type srv6-sid-value;
```

```
      description
        "SRV6 sid value.";
    }

    leaf neighbor-router-id {
       type yang:dotted-quad;
       description
         "Neighbor router ID.This is only
          used on LAN adjacencies.";
    }

    uses srv6-sid-structures;
  }
}

grouping srv6-locactor-tlvs {
  description
    "This group defines srv6 locator tlv.";
  container srv6-locactor-tlv {
    description
      "This contains a srv6 locator tlv.";
    leaf route-type {
      type identityref {
        base LOCATOR-ROUTE-TYPE;
      }
      description
        "The type of the locator route";
    }

    leaf algorithm {
      type uint8;
      description
        "Associated algorithm.";
    }

    leaf locator-length {
      type uint8;
      description
        "Carries the length of the Locator
         prefix as number of bits (1-128)";
    }

    leaf flags {
      type bits {
        bit n-flag {
          position 0;
          description
            "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";
        }
        bit a-flag {
          position 1;
          description
            "When the Locator is configured as anycast, the A bit
             SHOULD be set.  Otherwise, this bit MUST be clear";
        }
      }
      description
        "Flags for srv6 locator tlv.";
    }

    leaf metric {
      type uint32;
      description
        "Metric value.";
    }

    leaf-list locator {
      type inet:ipv6-address-no-zone;
      description
        "Advertised SRV6 locator.";
    }
    uses srv6-end-sids;
  }
}

/* Cfg */
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 augment OSPFv3 routing protocol when used";
  }
  description
    "This augments OSPFv3 protocol configuration
     with SRv6.";
  container srv6{
    leaf enable{
      type boolean;
      default "false";
      description
        "Enables SRv6 protocol extensions.";
    }
    leaf default-locator {
```

```
    type boolean;
    default "false";
    description
      "Enable OSPFv3 segment-routing IPv6 with default Locator.";
  }
  leaf-list locator-name {
    when "not(../default-locator='true')" {
      description
        "Only applies to non default locator.";
    }
    type leafref {
      path "/rt:routing/sr:segment-routing/srv6:srv6"
         + "/srv6:locators/srv6:locator/srv6:name";
    }
    description
      "Enable OSPFv3 segment-routing IPv6 with specified Locator.";
  }
  leaf persistent-end-x-sid{
    type boolean;
    default "false";
    description
      "Enable the persistent nature of End.X sid";
  }
  description
    "Configuration about OSPFv3 segment-routing IPv6.";
}

container micro-loop-avoidance {
  leaf srv6-enable {
    type boolean;
    default "false";
    description
      "Enable SRv6 avoid-microloop.Depend on SR IPv6 Enable.";
  }

  leaf srv6-rib-update-delay {
    type uint16 {
      range "1000..10000";
    }
    units "ms";
    default "5000";
    description
      "Set the route delivery delay for SRv6 avoid-microloop.
       Depend on SR IPv6 Enable.";
  }

  description
    "Enable OSPFv3 avoid-microloop.";
}
```

```
}

augment "/rt:routing/" +
        "rt:control-plane-protocols/rt:control-plane-protocol"+
        "/ospf:ospf/ospf:fast-reroute"{
  when "/rt:routing/rt:control-plane-protocols/"+
       "rt:control-plane-protocol/rt:type = 'ospf:ospfv3'"{
    description
      "This augment OSPFv3 routing protocol when used";
  }
  description
    "This augments OSPFv3 IP FRR with IPV6 TILFA.";

  container srv6-ti-lfa {
    if-feature srv6-ti-lfa;
    leaf enable {
      type boolean;
      description
        "Enables SRv6 TI-LFA computation.";
    }
    description
      "SRv6 TILFA configuration.";
  }
}

/* Database */
augment "/rt:routing/"
         + "rt:control-plane-protocols/rt:control-plane-protocol/"
         + "ospf:ospf/ospf:database/"
         + "ospf:as-scope-lsa-type/ospf:as-scope-lsas/"
         + "ospf:as-scope-lsa/ospf:version/ospf:ospfv3/"
         + "ospf:ospfv3/ospf:body" {
  when "/rt:routing/rt:control-plane-protocols/"+
       "rt:control-plane-protocol/rt:type = 'ospf:ospfv3'" {
    description
      "This augment OSPFv3 routing protocol when used";
  }
  description
    "This augments OSPFv3 protocol router capability.";
  container srv6-locator {
    description
      "SRv6 Locator LSA.";
    uses srv6-locactor-tlvs;
  }
}

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

```
          + "ospf:as-scope-lsa-type/ospf:as-scope-lsas/"
          + "ospf:as-scope-lsa/ospf:version/ospf:ospfv3/"
          + "ospf:ospfv3/ospf:body/ospf:router-information" {
    when "/rt:routing/rt:control-plane-protocols/"+
         "rt:control-plane-protocol/rt:type = 'ospf:ospfv3'" {
      description
        "This augment OSPFv3 routing protocol when used";
    }
    description
      "This augments OSPFv3 protocol router capability.";
    uses srv6-capabilities;
    uses srv6-msds;
  }

  augment "/rt:routing/rt:control-plane-protocols"
        + "/rt:control-plane-protocol/ospf:ospf/ospf:areas/ospf:area"
        + "/ospf:database/ospf:area-scope-lsa-type"
        + "/ospf:area-scope-lsas"
        + "/ospf:area-scope-lsa/ospf:version/ospf:ospfv3"
        + "/ospf:ospfv3/ospf:body/ospfv3-e-lsa:e-router/"
        + "ospfv3-e-lsa:e-router-tlvs/"
        + "ospfv3-e-lsa:link-tlv" {
    when "/rt:routing/rt:control-plane-protocols/"+
         "rt:control-plane-protocol/rt:type = 'ospf:ospfv3'" {
    }
    description
      "This augments OSPFv3 protocol neighbor.";
    uses srv6-endx-sids;
  }

  /* Notifications */
}

<CODE ENDS>
```

5. Security Considerations

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

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or

RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

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:

*srv6

*micro-loop-avoidance

*srv6-ti-lfa

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:

*srv6-locator

*srv6-capability

*srv6-msd

*srv6-endx-sid

6. Contributors

Qin Wu
Huawei
Email: bill.wu@huawei.com

7. Acknowledgements

TBD.

8. 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-ospfv3-srv6
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-ospfv3-srv6
namespace: urn:ietf:params:xml:ns:yang:ietf-ospfv3-srv6
prefix: ospfv3-srv6
reference: RFC XXXX

9. Normative References

[I-D.ietf-lsr-ospfv3-extended-lsa-yang] Lindem, A., Palani, S., and Y. Qu, "YANG Model for OSPFv3 Extended LSAs", Work in Progress, Internet-Draft, draft-ietf-lsr-ospfv3-extended-lsa-yang-08, 28 March 2021, <https://www.ietf.org/archive/id/draft-ietf-lsr-ospfv3-extended-lsa-yang-08.txt>.

[I-D.ietf-lsr-ospfv3-srv6-extensions] Li, Z., Hu, Z., Cheng, D., Talaulikar, K., and P. Psenak, "OSPFv3 Extensions for SRv6", Work in Progress, Internet-Draft, draft-ietf-lsr-ospfv3-srv6-extensions-02, 15 February 2021, <https://www.ietf.org/archive/id/draft-ietf-lsr-ospfv3-srv6-extensions-02.txt>.

[I-D.ietf-ospf-yang] Yeung, D., Qu, Y., Zhang, J., Chen, I., and A. Lindem, "YANG Data Model for OSPF Protocol", Work in Progress, Internet-Draft, draft-ietf-ospf-yang-29, 17 October 2019, <https://www.ietf.org/archive/id/draft-ietf-ospf-yang-29.txt>.

[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", Work in Progress, Internet-Draft, draft-ietf-rtgwg-segment-routing-ti-lfa-07, 29 June 2021, <https://www.ietf.org/archive/id/draft-ietf-rtgwg-segment-routing-ti-lfa-07.txt>.

[I-D.ietf-spring-srv6-yang]
Raza, K., Agarwal, S., Liu, X., Hu, Z., Hussain, I., Shah, H., Voyer, D., Matsushima, S., Horiba, K., AbdelSalam, A., and J. Rajamanickam, "YANG Data Model for SRv6 Base and Static", Work in Progress, Internet-Draft, draft-ietf-spring-srv6-yang-00, 9 September 2020,

<https://www.ietf.org/archive/id/draft-ietf-spring-srv6-yang-00.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>.

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

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

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

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

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

[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>.

Authors' Addresses

Zhibo Hu
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing
100095
China

Email: huzhibo@huawei.com

Xuesong Geng
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing
100095
China

Email: gengxuesong@huawei.com

Kamran Raza
Cisco Systems, Inc.
2000 Innovation Drive Kanata, ON K2K-3E8 CA

Email: skraza@cisco.com

Yingzhen Qu
Futurewei Technologies
2330 Central Express Way
Santa Clara, 950950
United States of America

Email: yingzhen.qu@futurewei.com

Acee Lindem
Cisco Systems
301 Midenhall Way
Cary, NC, 27513
United States of America

Email: acee@cisco.com