

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 25, 2019

Z. Hu
Huawei Technologies
K. Raza
Cisco Systems, Inc.
Y. Qu
J. Dong
Huawei Technologies
October 22, 2018

YANG Data Model for OSPF SRv6
draft-hu-lsr-ospf-srv6-yang-00

Abstract

This document defines a YANG data model that can be used to configure and manage OSPF SRv6 [[I-D.li-ospf-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 April 25, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents

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

Table of Contents

1.	Overview	2
2.	OSPF SRv6	2
3.	OSPF SRv6 configuration	4
3.1.	SRv6 activation	4
3.2.	IP Fast reroute	4
4.	OSPF SRv6 YANG Module	4
5.	Security Considerations	15
6.	Contributors	15
7.	Acknowledgements	15
8.	IANA Considerations	15
9.	References	16
	Authors' Addresses	17

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

[2.](#) OSPF SRv6

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

The OSPF SRv6 YANG module requires support of OSPF base model [[I-D.ietf-ospf-yang](#)] which defines basic OSPF configuration and state.

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

```
module: ietf-ospf-srv6
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/ospf:ospf:
      +--rw srv6-cfg
        +--rw enable?                boolean
        +--rw persistent-end-x-sid?  boolean
      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 srv6-ti-lfa {srv6-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:ospfv3
          /ospf:ospfv3/ospf:body/ospf:router-information:
            +--ro srv6-capability
              +--ro flags?            bits
              +--ro srv6-msd-value
                +--ro max-segments-left?  uint8
                +--ro max-end-pop?        uint8
                +--ro max-t-insert?       uint8
                +--ro max-t-encap?        uint8
                +--ro max-end-d?          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:ospfv3
          /ospf:ospfv3/ospf:body/ospfv3-e-lsa:e-router/
          /ospfv3-e-lsa:e-router-tlvs/ospfv3-e-lsa:link-tlv
          /ospfv3-e-lsa:sub-tlvs:
```

```

    +--ro srv6-endx-sids-sub-tlv
      +--ro endx-sid* [sid]
        +--ro endpoint-func
          | +--ro fuc-flags?          uint8
          | +--ro endpoint-func?      identityref
          | +--ro undefined-endpoint-func? uint16
        +--ro sid-flags?             uint8
        +--ro sid-size?              uint8
        +--ro sid                    srv6-sid-value
        +--ro neighbor-router-id?    yang:dotted-quad
augment /rt:routing/rt:control-plane-protocols
       /rt:control-plane-protocol/ospf:ospf/ospf:areas

```

```

    /ospf:area/ospf:interfaces/ospf:interface
    /ospf:database/ospf:link-scope-lsa-type
    /ospf:link-scope-lsas/ospf:link-scope-lsa
    /ospf:version/ospf:ospfv3/ospf:ospfv3/ospf:body
/ospf:router-information:
  +--ro srv6-node-sids
    +--ro srv6-node-sids* [sid]
      +--ro endpoint-func
        | +--ro fuc-flags?          uint8
        | +--ro endpoint-func?      identityref
        | +--ro undefined-endpoint-func? uint16
      +--ro sid-flags?             uint8
      +--ro sid-size?              uint8
      +--ro sid                    srv6-sid-value
      +--ro neighbor-router-id?    yang:dotted-quad

```

[3. OSPF SRv6 configuration](#)

[3.1. SRv6 activation](#)

Activation of OSPF 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.

[3.2. IP Fast reroute](#)

OSPF SRv6 model augments the fast-reroute container under interface. It brings the ability to activate ipv6 TI-LFA (topology independent

LFA).

4. OSPF SRv6 YANG Module

```
<CODE BEGINS> file "ietf-ospf-srv6@2018-10-11.yang"
module ietf-ospf-srv6 {
  namespace "urn:ietf:params:xml:ns:"
    + "yang:ietf-ospf-srv6";
  prefix ospf-srv6;

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

  import ietf-routing {
    prefix "rt";
  }

  import ietf-ospfv3-extended-lsa {
```

Hu, et al.

Expires April 25, 2019

[Page 4]

Internet-Draft

YANG Data Model for OSPF SRv6

October 2018

```
    prefix "ospfv3-e-lsa";
  }

  import ietf-ospf {
    prefix "ospf";
  }

  import iana-routing-types {
    prefix "iana-rt-types";
  }

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

  organization
    "IETF OSPF - OSPF Working Group";

  contact
    "WG List: <mailto:spring@ietf.org>
      Zhibo Hu
      <mailto:huzhibo@huawei.com>";
```

```

        Jiajia Dong
        <mailto:dongjiajia@huawei.com>;
";

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

revision 2018-10-11 {
    description
        "Initial revision.";
    reference "RFC XXXX";
}

/* 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 .";
}

/* 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-msd {
  description
    "means to advertise to advertise node/link specific values for
    Maxium Sid Depths(MSD) of various types";
  container v6-max-value-srh {
    description
      "Maximum SRv6 SID Depths.";
    leaf max-segments-left {
      type uint8;
      description
        "The maximum value of 'SL' field in the SRH of a received packet.";
    }
    leaf max-end-pop {
      type uint8;
      description
        "The maximum number of SIDS in the top SRH in an SRH stack to which
        the router can apply 'PSP' or 'USP'.";
    }
    leaf max-t-insert {
      type uint8;
      description
        "The maximum number of SIDs can be inserted as port of the 'T.insert'
        behavior.";
    }
    leaf max-t-encap {
      type uint8;
      description
        "The maximum number of SIDs can be included as part of the 'T.Encap'
        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 e-flag {
      position 0;
      description
        "If set, then router is able to
        apply T.Encap operation as specified in
        [filsfils-spring-srv6-network-programming].";
    }
    bit o-flag {
      position 1;
      description
        "If set, then the router is able to
        use of the O-bit in the Segment Routing Header(SRH)
        as defined in [draft-ietf-6man-segment-routing-header].";
    }
  }
  description
    "Flags.";
}

container srv6-msd-value {
  description
    "Maximum SRv6 SID Depths.";
  leaf max-segments-left {
    type uint8;
    description
      "The maximum value of 'SL' field in the SRH of a received packet.";
  }
  leaf max-end-pop {
    type uint8;
    description
      "The maximum number of SIDS in the top SRH in an SRH stack to which
      the router can apply 'PSP' or 'USP'.";
  }
  leaf max-t-insert {
    type uint8;
    description
      "The maximum number of SIDs can be inserted as port of the 'T.insert'
      behavior.";
  }
  leaf max-t-encap {
    type uint8;
    description
      "The maximum number of SIDs can be included as part of the 'T.Encap'
      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-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 ospf-srv6:SRV6_END_FUNC_TYPE;
      }
      description
        "The endpoint function.";
    }
    leaf undefined-endpoint-func {
      type uint16;
      description
        "Unknown endpoint func value.";
    }
  }
}

grouping 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.";
}
// sub-sub-tlvs not yet defined
}

grouping srv6-node-sid {
    description
        "This group defines srv6 node sid tlv.";
    uses srv6-endpoint-func;

    leaf sid-flags {
        type uint8;
        description
            "NO sid flags are currently being defined.";
    }

    leaf sid-size {
        type uint8;
        description
            "Number of bits in the SID field";
    }

    leaf 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.";
    }

    // sub-tlvs
}

grouping srv6-endx-sid {
    description
        "SRv6 sid associated with an adjacency.";

    uses srv6-endpoint-func;

    leaf sid-flags {
        type uint8;
        description

```

```

        "NO sid flags are currently being defined.";
    }

    leaf sid-size {
        type uint8;
        description
            "Number of bits in the SID field";
    }

    leaf 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.";
    }

    // sub-tlvs
}

```

```

/* 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 OSPF routing protocol when used";
    }
    description
        "This augments OSPF protocol configuration
        with SRv6.";

    container srv6-cfg{
        leaf enable{
            type boolean;
            default "false";
            description
                "Enables SRv6
                protocol extensions.";
        }
    }
}

```

```

    leaf persistent-end-x-sid{
        type boolean;
        default "false";
        description
            "Enable the persistent nature of End.X sid";
    }
    description
        "Configuration about OSPF segment-routing IPv6.";
}
}

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:routing/rt:control-plane-protocols/" +
        "rt:control-plane-protocol/rt:type = 'ospf:ospfv3'"{
        description

```

```

        "This augment OSPF routing protocol when used";
    }
    description
        "This augments OSPF IPFRR 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: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: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;
}

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/"
    + "ospfv3-e-lsa:sub-tlvs" {
when "/rt:routing/rt:control-plane-protocols/"+
    "rt:control-plane-protocol/rt:type = 'ospf:ospfv3'" {
}
description
    "This augments OSPFv3 protocol neighbor.";
container srv6-endx-sids-sub-tlv {
    description
        "This defines svr6 end-x sids for the adjacency.";
    list endx-sid {
        key "sid";
        uses srv6-endx-sid;
        description
            "List of end-x sids.";
    }
}
}
}

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:routing/rt:control-plane-protocols/"+
    "rt:control-plane-protocol/rt:type = 'ospf:ospfv3'" {
    description
        "This augment OSPFv3 routing protocol when used.";
}
description
    "This augment OSPFv3 routing protocol when used";

```

```

container srv6-node-sids {
    description
        "This defines srv6 node sid tlvs.";
    list srv6-node-sids {

```



```

        key "sid";
        uses srv6-node-sid;
        description
            "List of srv6 node sids.";
    }
}
}
/* Notifications */
}
<CODE ENDS>

```

5. Security Considerations

Configuration and state data defined in this document are designed to be accessed via the NETCONF protocol [[RFC6241](#)].

As OSPF is an IGP protocol (critical piece of the network), ensuring stability and security of the protocol is mandatory for the network service.

Authors recommends to implement NETCONF access control model ([[RFC6536](#)]) to restrict access to all or part of the configuration to specific users.

6. Contributors

TBD.

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-ospf-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-isis-srv6
namespace: urn:ietf:params:xml:ns:yang:ietf-ospf-srv6 prefix: ospf-srv6
reference: RFC XXXX

9. 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-17](#) (work in progress), September 2018.

[I-D.li-ospf-ospfv3-srv6-extensions]

Li, Z., Hu, Z., Cheng, D., Talaulikar, K., and P. Psenak,
"OSPFv3 Extensions for SRv6", [draft-li-ospf-ospfv3-srv6-extensions-02](#) (work in progress), September 2018.

[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>>.

Internet-Draft

YANG Data Model for OSPF SRv6

October 2018

Authors' Addresses

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

Email: huzhibo@huawei.com

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

Email: skraza@cisco.com

Yingzhen Qu
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: yingzhen.qu@huawei.com

Jiajia Dong
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: dongjiajia@huawei.com

Hu, et al.

Expires April 25, 2019

[Page 17]