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

Z. Hu
Huawei Technologies
K. Raza
Cisco Systems, Inc.
Y. Qu
J. Dong
Huawei Technologies
November 4, 2019

YANG Data Model for OSPF SRv6
draft-hu-lsr-ospf-srv6-yang-01

Abstract

This document defines a YANG data model that can be used to configure and manage OSPFv3 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 May 7, 2020.

Copyright Notice

Table of Contents

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

2. OSPFv3 SRv6

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.

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-cfg
        |  +--rw enable?                 boolean
        |  +--rw default-locator?        boolean
        |  +--rw locator-name?           -> /rt:routing/srv6:srv6
        |  |                                /locators/locator/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 v6-max-value-srh
           +--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: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 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
```

3. OSPFv3 SRv6 configuration

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.

3.2. Locator setting

The basic SRv6 module defines the related locator leafs. When the OSPFv3 SRv6 module is enabled, set the locator by using the following strategy: firstly, it is reasonable to check whether the default locator is used, if not, to use the specified locator. The strategy is realized by adding the leaf "default-locator", "locator-name" .

3.3. IP Fast reroute

OSPFv3 SRv6 model augments the fast-reroute container under interface. It brings the ability to activate ipv6 TI-LFA (topology independent LFA).

3.4. Microloop avoidance

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

4. OSPFv3 SRv6 YANG Module

```
<CODE BEGINS> file "ietf-ospfv3-srv6@2019-11-04.yang"
module ietf-ospfv3-srv6 {
  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 iana-routing-types {
    prefix "iana-rt-types";
  }

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

  organization
   "IETF LSR Working Group";

  contact
    "WG List:  &lt;mailto:spring@ietf.org&gt;
        Zhibo Hu
          &lt;mailto:huzhibo@huawei.com&gt;
```

```
       Jiajia Dong
         &lt;mailto:dongjiajia@huawei.com&gt;
    ";

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

  revision 2019-11-04 {
    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.";
  }

  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-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 recevied 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 o-flag {
            position 1;
            description
```



```
              "If set, then router is capable of supporting SRH O-bit,
               as specified in [I-D.ali-spring-srv6-oam]";
          }
        }
        description
          "Flags.";
      }
    }
  }

  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-cfg{
      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 locator-name {
        when "not(../default-locator='true')";
        type leafref {
          path "/rt: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 I
      }
```

```
      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-msd;
  }

  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 "/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

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-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. 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.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-05 (work in progress), August 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>.

[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>.

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