Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 14, 2021

Z. Hu
Huawei Technologies
D. Ye
Cisco Systems
Y. Qu
Futurewei Technologies
X. Geng
Huawei Technologies

July 13, 2020

YANG Data Model for IS-IS SRv6
draft-hu-isis-srv6-yang-04

Abstract

This document defines a YANG data model that can be used to configure and manage IS-IS SRv6 [I-D.ietf-lsr-isis-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 January 14, 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
2. IS-IS SRv6 2
3. IS-IS SRv6 configuration 5
3.1. SRv6 activation 5
3.2. Locator setting 5
3.3. IP Fast reroute 5
3.4. Microloop avoidance 6
4. IS-IS SRv6 YANG Module 6
5. Security Considerations 19
6. Contributors 20
7. Acknowledgements 20
8. IANA Considerations 20
9. References 20
Authors' Addresses 21

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 IS-IS SRv6 and it is an augmentation to the IS- IS YANG data model.

2. IS-IS SRv6

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

The IS-IS SRv6 YANG module requires support for the base srv6 module[I-D.raza-spring-srv6-yang], which defines the global srv6 configuration independent of any specific routing protocol configuration, and support of IS-IS base model [I-D.ietf-isis-yang-isis-cfg] which defines basic IS-IS configuration and state.

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

```
module: ietf-isis-srv6
module: ietf-isis-srv6
  augment /rt:routing/rt:control-plane-protocols
          /rt:control-plane-protocol/isis:isis:
    +--rw srv6-cfg
    |  +--rw enable?                boolean
    |  +--rw default-locator?       boolean
    |  +--rw locator-name*          -> /rt:routing/sr:segment-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/isis:isis/isis:fast-reroute:
    +--rw srv6-ti-lfa {srv6-ti-lfa}?
       +--rw enable?   boolean
  augment /rt:routing/rt:control-plane-protocols
          /rt:control-plane-protocol/isis:isis/isis:database
          /isis:levels/isis:lsp/isis:router-capabilities:
    +--ro v6-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/isis:isis/isis:database
          /isis:levels/isis:lsp/isis:extended-is-neighbor/isis:neighbor:
    +--ro srv6-adjacency-sids
       +--ro end-x-sid* [sid]
          +--ro func-flags?           bits
          +--ro algorithm?            uint8
          +--ro weight?               uint8
          +--ro endpoint-func
          |  +--ro flags?                     uint8
          |  +--ro endpoint-func?             identityref
```

```
          |  +--ro undefined-endpoint-func?   uint16
          +--ro sid                   srv6-sid-value
          +--ro neighbor-id?          isis:system-id
          +--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/isis:isis/isis:database
          /isis:levels/isis:lsp/isis:mt-is-neighbor/isis:neighbor:
    +--ro srv6-adjacency-sids
       +--ro end-x-sid* [sid]
          +--ro func-flags?           bits
          +--ro algorithm?            uint8
          +--ro weight?               uint8
          +--ro endpoint-func
          |  +--ro flags?                     uint8
          |  +--ro endpoint-func?             identityref
          |  +--ro undefined-endpoint-func?   uint16
          +--ro sid                   srv6-sid-value
          +--ro neighbor-id?          isis:system-id
          +--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/isis:isis/isis:database
          /isis:levels/isis:lsp:
    +--ro srv6-locators
       +--ro locator* [locator]
          +--ro mt-id?                   uint16
          +--ro flags?                   bits
          +--ro metric?                  uint32
          +--ro algorithm?               uint8
          +--ro loc-size?                uint8
          +--ro locator                  inet:ipv6-address-no-zone
          +--ro srv6-end-sids
          |  +--ro end-sid* [sid]
          |     +--ro flags?                uint8
          |     +--ro endpoint-func
          |     |  +--ro 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
          +--ro external-prefix-flag?    boolean
          +--ro readvertisement-flag?    boolean
          +--ro node-flag?               boolean
          +--ro ipv4-source-router-id?   inet:ipv4-address
          +--ro ipv6-source-router-id?   inet:ipv6-address
  augment /rt:routing/rt:control-plane-protocols
          /rt:control-plane-protocol/isis:isis/isis:interfaces
          /isis:interface/isis:adjacencies/isis:adjacency:
    +--ro end-x-sid* [value]
       +--ro value                   srv6-sid-value
       +--ro weight?                 uint8
       +--ro protection-requested?   boolean
       +--ro persistent?             boolean
       +--ro algorithm?              uint8
       +--ro endpoint-func
          +--ro flags?                     uint8
          +--ro endpoint-func?             identityref
          +--ro undefined-endpoint-func?   uint16
```

3. IS-IS SRv6 configuration

3.1. SRv6 activation

Activation of IS-IS 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 IS-IS 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

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

3.4. Microloop avoidance

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

4. IS-IS SRv6 YANG Module

```
<CODE BEGINS> file "ietf-isis-srv6@2020-07-13.yang"
module ietf-isis-srv6 {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:"
          + "yang:ietf-isis-srv6";
  prefix isis-srv6;

  import ietf-routing {
    prefix "rt";
  }

  import ietf-isis {
    prefix "isis";
  }

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

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

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

  import ietf-segment-routing {
    prefix sr;
    reference "draft-ietf-spring-sr-yang";
  }

  organization
   "IETF LSR Working Group";

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

```
                 <mailto:daye@cisco.com>
      Author:    Yingzhen Qu
                 <mailto:yingzhen.qu@futurewei.com>
      Author:    Jiajia Dong
                 <mailto:dongjiajia@huawei.com>
      ";
    description
      "The YANG module defines a generic configuration model for
      Segment IPV6 routing ISIS extensions common across all of
      the vendor implementations.";

    revision 2020-07-13 {
      description
          "Initial revision.";
      reference "draft-ietf-lsr-isis-srv6-extensions-08";
    }

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

    identity SRV6_END_FUNC_PSP_USP_USD {
      base "SRV6_END_FUNC_TYPE";
      description
      "End (May support PSP, USP, USD).";
    }

    identity SRV6_END_X_FUNC_PSP_USP_USD {
      base "SRV6_END_FUNC_TYPE";
      description
      "End.X(May support PSP, USP, USD)";
    }

    identity SRV6_END_T_FUNC_PSP_USP_USD {
      base "SRV6_END_FUNC_TYPE";
      description
      "END (May support PSP, USP, USD)";
    }

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

    identity SRV6_END_FUNC_DX4 {
```

```
      base "SRV6_END_FUNC_TYPE";
      description
      "End.DX4.";
    }

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

    identity SRV6_END_FUNC_DT4 {
      base "SRV6_END_FUNC_TYPE";
      description
      "End.DT4.";
    }

    identity SRV6_END_FUNC_DT64 {
      base "SRV6_END_FUNC_TYPE";
      description
      "End.DT64.";
    }

    identity SRV6_END_FUNC_OP {
      base "SRV6_END_FUNC_TYPE";
      description
      "END.OP .";
    }

    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-msds {
      description
        "means to advertise to advertise node/link specific
         values for Maxium 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-sid-structures {
      description
        "This group defines SRv6 SID Structure sub-sub-TLV.";
      container srv6-sid-structure {
        description
          "SRv6 SID Structure sub-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-capability {
      description
      "SRV6 capability grouping.";
      container v6-capability {
        description
          "SRv6 capability.";
        leaf flags {
          type bits {
            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.";
        }
      }
    }

    grouping srv6-endpoint-func {
```

```
       description
        "This group defines srv6 endpoint function";
      container endpoint-func {
        description
          "Srv6 Endpoint function Descriptor.";
        leaf flags {
          type uint8;
          description
            "No flags are currently being defined.";
        }
        leaf endpoint-func {
          type identityref {
           base isis-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
      uses srv6-sid-structures;
    }

    grouping srv6-locator {
      description
        "This group defines srv6 locator tlv.";
```

```
      leaf mt-id {
        type uint16 {
          range "0..4095";
        }
        description
          "Multitopology Identifier as defined in [RFC5120].";
      }
      leaf flags {
        type bits {
          bit d-flag {
            position 0;
            description
              "When the locator is leaked from level-2 to leve-1,
               the d-flag must be set.";
          }
        }
        description
          "Flags for srv6 locator tlv.";
      }

      leaf metric {
        type uint32;
        description
          "Metric value.";
      }
      leaf algorithm {
        type uint8;
        description
        "Associated algorithm.";
      }

      leaf loc-size {
        type uint8;
        description
          "Number of bits in the locator field.";
      }
      leaf locator {
        type inet:ipv6-address-no-zone;
        description
          "Advertised SRV6 locator.";
      }
      container srv6-end-sids {
        description
          "This contains list of srv6 end sids.";
        list end-sid {
          key "sid";
          description
            "List of SRV6 SRv6 Segment Identifiers (SID)
```

```
             with Endpoint functions.";
          uses srv6-end-sid;
        }
      }
      uses isis:prefix-reachability-attributes;
      uses isis:prefix-ipv4-source-router-id;
      uses isis:prefix-ipv6-source-router-id;
    }

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

      leaf func-flags {
        type bits {
          bit b-flag {
            position 0;
            description
              "Backup flag. If set, the End.X sid is
               eligible for protection.";
          }

          bit s-flag {
            position 1;
            description
              "Set flag. When set, 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. When set, the End.X sid is persistently
             allocated, i.e., the End.x sid value remains consistent
             across router restart and/or interface flap.";
          }
        }
        description
            "Flags for srv6 end x sid.";
      }

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

```
      leaf weight {
        type uint8;
        description
        "The value represents the weight of the End.X sid
         for the purpose of load balancing.";
      }

      uses srv6-endpoint-func;

      leaf sid {
        type srv6-sid-value;
        description
          "SRV6 sid value.";
      }

      leaf neighbor-id {
        type isis:system-id;
        description
          "Describes the system ID of the neighbor
           associated with the SID value. This is only
           used on LAN adjacencies.";
      }

      // sub-sub-tlvs
      uses srv6-sid-structures;
    }

    grouping srv6-adjacency-state {
      description
       "This group will extend adjacency state.";
      list end-x-sid {
        key value;
        config false;
        leaf value {
          type srv6-sid-value;
          description
           "Value of the Adj-SID.";
        }
        leaf weight {
          type uint8;
          description
           "Weight associated with
           the End.X SID.";
        }
        leaf protection-requested {
          type boolean;
          description
           "Set to True if the End.X SID
```

```
           must be protected.";
        }
        leaf persistent {
          type boolean;
          description
            "Persistent flag. When set, the End.X sid is persistently
             allocated, i.e., the End.X sid value remains consistent
             across router restart and/or interface flap.";
        }
        leaf algorithm {
          type uint8;
          description
          "Associated algorithm.";
        }
        uses srv6-endpoint-func;

        description
         "List of End.X Segment IDs.";
      }
    }
    /* Cfg */
    augment "/rt:routing/" +
            "rt:control-plane-protocols/rt:control-plane-protocol"+
            "/isis:isis" {
      when "/rt:routing/rt:control-plane-protocols/"+
           "rt:control-plane-protocol/rt:type = 'isis:isis'" {
        description
          "This augment ISIS routing protocol when used";
      }
      description
        "This augments ISIS protocol configuration
         with SRv6.";

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

        leaf default-locator {
          type boolean;
          default "false";
          description
            "Enable ISIS segment-routing IPv6 with default Locator.";
        }
```

```
        leaf-list locator-name {
          when "derived-from(../default-locator, 'false')" {
            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 ISIS 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 ISIS 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 IS-IS avoid-microloop.";
      }
    }

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

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

    /* Operational states */
    augment "/rt:routing/" +
          "rt:control-plane-protocols/rt:control-plane-protocol"+
          "/isis:isis/isis:database/isis:levels/isis:lsp"+
          "/isis:router-capabilities" {
       when "/rt:routing/rt:control-plane-protocols/"+
          "rt:control-plane-protocol/rt:type = 'isis:isis'" {
        description
          "This augment ISIS routing protocol when used";
      }
      description
       "This augments ISIS protocol router capability.";
        uses srv6-capability;
        uses srv6-msds;
    }

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

```
      }
      description
       "This augments ISIS protocol neighbor.";
      container srv6-adjacency-sids {
        description
          "This defines svr6 end-x sids for the adjacency.";
        list end-x-sid {
          key "sid";
          uses srv6-adjacency-sid;
          description
             "List of end-x sids.";
        }
      }
    }
    augment "/rt:routing/" +
          "rt:control-plane-protocols/rt:control-plane-protocol"+
          "/isis:isis/isis:database/isis:levels/isis:lsp"+
          "/isis:mt-is-neighbor/isis:neighbor" {
              when "/rt:routing/rt:control-plane-protocols/"+
          "rt:control-plane-protocol/rt:type = 'isis:isis'" {
        description
          "This augment ISIS routing protocol when used.";
      }
      description
        "This augments ISIS protocol neighbor.";
      container srv6-adjacency-sids {
         description
          "This defines svr6 end-x sids for the adjacency.";
        list end-x-sid {
          key "sid";
          uses srv6-adjacency-sid;
           description
             "List of end-x sids.";
        }
      }
    }

    augment "/rt:routing/" +
          "rt:control-plane-protocols/rt:control-plane-protocol"+
          "/isis:isis/isis:database/isis:levels/isis:lsp" {
          when "/rt:routing/rt:control-plane-protocols/"+
          "rt:control-plane-protocol/rt:type = 'isis:isis'" {
        description
        "This augment ISIS routing protocol when used.";
      }
      description
       "This augments ISIS protocol LSDB.";
      container srv6-locators {
```

```
        description
           "This defines srv6 locator tlvs.";
        list locator {
          key "locator";
          uses srv6-locator;
          description
            "List of srv6 locators.";
        }
      }
    }

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

      uses  srv6-adjacency-state;
    }

    /* 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 IS-IS 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

Jiajia Dong
Huawei
China

Email:dongjiajia@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-isis-srv6
Registrant Contact: IS-IS WG
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-isis-srv6 prefix: isis-srv6
reference: RFC XXXX
```

9. References

[I-D.ietf-isis-yang-isis-cfg]
Litkowski, S., Yeung, D., Lindem, A., Zhang, Z., and L. Lhotka, "YANG Data Model for IS-IS Protocol", draft-ietf-isis-yang-isis-cfg-42 (work in progress), October 2019.

[I-D.ietf-lsr-isis-srv6-extensions]
Psenak, P., Filsfils, C., Bashandy, A., Decraene, B., and Z. Hu, "IS-IS Extension to Support Segment Routing over IPv6 Dataplane", draft-ietf-lsr-isis-srv6-extensions-08 (work in progress), April 2020.

[I-D.raza-spring-srv6-yang]
Raza, K., Agarwal, S., Liu, X., Hu, Z., Hussain, I., Shah, H., Voyer, D., Elmalky, H., Matsushima, S., Horiba, K., Abdelsalam, A., and J. Rajamanickam, "YANG Data Model for SRv6 Base and Static", draft-raza-spring-srv6-yang-05 (work in progress), October 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

Dan Ye
Cisco Systems
170 W. Tasman Dr. San Jose,
California 95134
USA

Email: daye@cisco.com

Yingzhen Qu
Futurewei Technologies
2330 Central Express Way
Santa Clara 950950
USA

Email: yingzhen.qu@futurewei.com

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

Email: gengxuesong@huawei.com