SPRING WG
Internet-Draft
Intended status: Standards Track
Expires: April 18, 2021

R. Chen
D. Zhao
ZTE Corporation

October 15, 2020

YANG data model for shorter srv6
draft-chen-spring-shorter-srv6-yang-01

Abstract

This document is to define the YANG data model for shorter srv6(Compressing SRv6).

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 18, 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. Introduction 2
2. Shorter SRv6 Types 2
3. Configuration 2
3.1. Shorter SRv6 base 2
3.2. Shorter SRv6 Static 3
4. YANG Module 6
4.1. Shorter SRv6 type 6
4.2. Shorter SRv6 Base 10
4.3. Shorter SRv6 Static 12
5. Security Considerations 22
6. IANA Considerations 22
7. Normative References 22
Authors' Addresses 23

1. Introduction

YANG[RFC6020]is a data definition language that was introduced to define the contents of a conceptual data store that allows networked devices to be managed using NETCONF [RFC6241]. This document defines a YANG data model for the configuration of shorter srv6.

[I-D.mirsky-6man-unified-id-sr] proposed an extension of SRH that enables the use of a shorter segment identifier, such as 32-bits Label format SID or 32-bits IP address format SID.

[I-D.ietf-spring-srv6-yang]introduces a YANG data model for base SRv6 and SRv6 Static application.

This document is to define the YANG data model for shorter srv6(Compressing SRv6)

2. Shorter SRv6 Types

The Shorter-SRv6-Types augment "ietf-srv6-types" [I-D.ietf-spring-srv6-yang], and defines new srv6-endpoint-type: SRv6 Endpoint behaviors identity type.

3. Configuration

3.1. Shorter SRv6 base

The Shorter-SRv6-base configuration augments the SRv6-base locator tree [I-D.ietf-spring-srv6-yang]"/rt:routing/rt:srv6", and defines new Parameters.

Following is a simplified graphical tree representation of the data model for Shorter SRv6 base configuration.

```
  module: ietf-shorter-srv6-base
        augment /rt:routing/rt:srv6:
         +--rw shorter-srv6
            +--rw node-capabilities
                   +--rw UET-support?            enumeration


     augment /rt:routing/rt:srv6/rt:locators/rt:locator/rt:prefix:
          +--rw lb-length     srv6-types:srv6-lb-len
```

3.2. Shorter SRv6 Static

The Shorter-SRv6-Static configuration augments the SRv6-base locator tree "/rt:routing/srv6:srv6/srv6:locators/srv6:locator/rt:prefix:", and defines new Parameters.

Following is a simplified graphical tree representation of the data model for Shorter SRv6 Static configuration.

```
      module: ietf-shorter-srv6-static
      augment /rt:routing/srv6:srv6/srv6:locators/srv6:locator:/srv6:static/
srv6:local-sids/srv6:sid:
         +--rw end_uet32
         +--rw end_psp_uet32
         +--rw end_usp_uet32
         +--rw end_usd_uet32
         +--rw end_psp_usp_uet32
         +--rw end_psp_usp_usd_uet32
         +--rw end.x_uet32
      |  +--rw protected?   boolean
      |  +--rw paths
      |     +--rw path* [path-index]
      |        +--rw path-index           uint8
      |        +--rw interface?           if:interface-ref
      |        +--rw next-hop?            inet:ipv6-address
      |        +--rw table?               srv6-types:table-id
      |        +--rw weight?              uint32
      |        +--rw role?                enumeration
      |        +--rw backup-path-index?   uint8
      |        +--rw sid-list
      |           +--rw out-sid* [sid]
      |              +--rw sid    srv6-types:srv6-sid
      +--rw end.x_psp_uet32
```

```
            |  +--rw protected?   boolean
        |  +--rw paths
        |     +--rw path* [path-index]
        |        +--rw path-index           uint8
        |        +--rw interface?           if:interface-ref
        |        +--rw next-hop?            inet:ipv6-address
        |        +--rw table?               srv6-types:table-id
        |        +--rw weight?              uint32
        |        +--rw role?                enumeration
        |        +--rw backup-path-index?   uint8
        |        +--rw sid-list
        |           +--rw out-sid* [sid]
        |              +--rw sid    srv6-types:srv6-sid
            +--rw end.x_usp_uet32
            |  +--rw protected?   boolean
        |  +--rw paths
        |     +--rw path* [path-index]
        |        +--rw path-index           uint8
        |        +--rw interface?           if:interface-ref
        |        +--rw next-hop?            inet:ipv6-address
        |        +--rw table?               srv6-types:table-id
        |        +--rw weight?              uint32
        |        +--rw role?                enumeration
        |        +--rw backup-path-index?   uint8
        |        +--rw sid-list
        |           +--rw out-sid* [sid]
        |              +--rw sid    srv6-types:srv6-sid
            +--rw end.x_usd_uet32    |  +--rw protected?   boolean
        |  +--rw paths
        |     +--rw path* [path-index]
        |        +--rw path-index           uint8
        |        +--rw interface?           if:interface-ref
        |        +--rw next-hop?            inet:ipv6-address
        |        +--rw table?               srv6-types:table-id
        |        +--rw weight?              uint32
        |        +--rw role?                enumeration
        |        +--rw backup-path-index?   uint8
        |        +--rw sid-list
        |           +--rw out-sid* [sid]
        |              +--rw sid    srv6-types:srv6-sid
        +--rw end.x_psp_usp_uet32
            |  +--rw protected?   boolean
        |  +--rw paths
        |     +--rw path* [path-index]
        |        +--rw path-index           uint8
        |        +--rw interface?           if:interface-ref
        |        +--rw next-hop?            inet:ipv6-address
        |        +--rw table?               srv6-types:table-id
```

```
        |        +--rw weight?              uint32
        |        +--rw role?                enumeration
        |        +--rw backup-path-index?   uint8
        |        +--rw sid-list
        |           +--rw out-sid* [sid]
        |              +--rw sid    srv6-types:srv6-sid
            +--rw end.x_psp_usd_uet32
            |  +--rw protected?   boolean
        |  +--rw paths
        |     +--rw path* [path-index]
        |        +--rw path-index           uint8
        |        +--rw interface?           if:interface-ref
        |        +--rw next-hop?            inet:ipv6-address
        |        +--rw table?               srv6-types:table-id
        |        +--rw weight?              uint32
        |        +--rw role?                enumeration
        |        +--rw backup-path-index?   uint8
        |        +--rw sid-list
        |           +--rw out-sid* [sid]
        |              +--rw sid    srv6-types:srv6-sid
            +--rw end.x_usp_usd_uet32
            |  +--rw protected?   boolean
        |  +--rw paths
        |     +--rw path* [path-index]
        |        +--rw path-index           uint8
        |        +--rw interface?           if:interface-ref
        |        +--rw next-hop?            inet:ipv6-address
        |        +--rw table?               srv6-types:table-id
        |        +--rw weight?              uint32
        |        +--rw role?                enumeration
        |        +--rw backup-path-index?   uint8
        |        +--rw sid-list
        |           +--rw out-sid* [sid]
        |              +--rw sid    srv6-types:srv6-sid
            +--rw end-x_psp_usp_usd_uet32
            |  +--rw protected?   boolean
        |  +--rw paths
        |     +--rw path* [path-index]
        |        +--rw path-index           uint8
        |        +--rw interface?           if:interface-ref
        |        +--rw next-hop?            inet:ipv6-address
        |        +--rw table?               srv6-types:table-id
        |        +--rw weight?              uint32
        |        +--rw role?                enumeration
        |        +--rw backup-path-index?   uint8
        |        +--rw sid-list
        |           +--rw out-sid* [sid]
        |              +--rw sid    srv6-types:srv6-sid
```

```
           +--rw end.t_uet32
           |  +--rw lookup-table-ipv6    srv6-types:table-id
           +--rw end.t_psp_uet32
           |  +--rw lookup-table-ipv6    srv6-types:table-id
           +--rw end.t_psp_usp_uet32
           |  +--rw lookup-table-ipv6    srv6-types:table-id
       +--rw end.t_usd_uet32
           |  +--rw lookup-table-ipv6    srv6-types:table-id
       +--rw end.t_psp_usd_uet32
           |  +--rw lookup-table-ipv6    srv6-types:table-id
       +--rw end.t_usp_usd_uet32
           |  +--rw lookup-table-ipv6    srv6-types:table-id
           +--rw end-t_psp_usp_usd_uet32
              +--rw lookup-table-ipv6    srv6-types:table-id
```

4. YANG Module

4.1. Shorter SRv6 type

The Shorter-SRv6-Types augment "ietf-srv6-types" [I-D.ietf-spring-srv6-yang], and defines new srv6-endpoint-type.

```
     <CODEBEGINS> file "ietf-shorter-srv6-type@2020-03-30.yang"

     module ietf-shorter-srv6-types {
     yang-version 1.1;

     namespace "urn:ietf:params:xml:ns:yang:ietf-shorter-srv6-types";
     prefix "shorter-srv6-types";

     import ietf-inet-types {
     prefix "inet";
             reference "RFC6991: Common YANG Data Types";
     }
 import ietf-srv6-types {
 prefix srv6-types;
 reference "RFC XXXX: YANG Data Model for SRv6";
 }

     organization
    "IETF SPRING - SPRING Working Group";
  contact
    "WG Web:   <http://tools.ietf.org/wg/spring/>
     WG List:  <mailto:spring@ietf.org>
     Editor: Ran Chen
                 <mailto:chen.ran@zte.com.cn>
         Editor: Detao Zhao
                 <mailto:zhao.detao@zte.com.cn>
```

```
                        ";
        description
         "The YANG module defines a generic configuration model for SFF.
                  Copyright (c) 2019 IETF Trust and the persons
          identified as authors of the code.  All rights reserved.

          Redistribution and use in source and binary forms, with or
          without modification, is permitted pursuant to, and subject
          to the license terms contained in, the Simplified BSD License
          set forth in Section 4.c of the IETF Trust's Legal Provisions
          Relating to IETF Documents
          (https://trustee.ietf.org/license-info).
          This version of this YANG module is part of RFC XXXX; see
          the RFC itself for full legal notices.";

       revision "2020-03-30"{
             description "Initial revision.";
         reference "RFC XXXX: YANG Data Model for Shorter SRv6 type.";
     }

          /* Identities */
     identity srv6-endpoint-type {
      description
        "Base identity from which specific SRv6 Endpoint types are
        derived.";
    }
   identity End_UET32 {
    base srv6-endpoint-type;
    description
          "End function (variant: UET32 only).";
    }

    identity End_PSP_UET32{
    base srv6-endpoint-type;
    description
          "End function (variant:PSP and UET32).";
    }

        identity End_USP_UET32{
    base srv6-endpoint-type;
    description
          "End function (variant:USP and UET32).";
    }
    identity End_PSP_USP_UET32{
    base srv6-endpoint-type;
    description
          "End function (variant:PSP, USP, and UET32).";
    }
```

```
        identity USD_UET32{
    base srv6-endpoint-type;
    description
          "End function (variant: USD, and UET32).";
    }
        identity End_PSP_USD_UET32{
    base srv6-endpoint-type;
    description
          "End function (variant:PSP, USD, and UET32).";
    }
        identity End_USP_USD_UET32{
    base srv6-endpoint-type;
    description
          "End function (variant:USP, USD, and UET32).";
    }
        identity End_PSP_USP_USD_UET32{
    base srv6-endpoint-type;
    description
          "End function (variant:PSP, USP, USD, and UET32).";
    }

        identity End.X_UET32 {
    base srv6-endpoint-type;
    description
          "Endpoint with cross-connect to an array
           of layer-3 adjacencies (variant: UET32).";
    }
    identity End.X_PSP_UET32 {
    base srv6-endpoint-type;
    description
          "Endpoint with cross-connect to an array
           of layer-3 adjacencies (variant:PSP and UET32).";
    }
        identity End.X_USP_UET32 {
    base srv6-endpoint-type;
    description
          "Endpoint with cross-connect to an array
           of layer-3 adjacencies (variant: USP and UET32).";
    }
        identity End.X_PSP_USP_UET32 {
    base srv6-endpoint-type;
    description
          "Endpoint with cross-connect to an array
           of layer-3 adjacencies (variant:PSP , USP and UET32).";
    }
        identity End.X_USD_UET32 {
    base srv6-endpoint-type;
    description
```

```
          "Endpoint with cross-connect to an array
           of layer-3 adjacencies (variant:USD and UET32).";
    }
        identity End.X_PSP_USD_UET32 {
    base srv6-endpoint-type;
    description
          "Endpoint with cross-connect to an array
           of layer-3 adjacencies (variant:PSP, USD and UET32).";
    }
        identity End.X_USP_USD_UET32 {
    base srv6-endpoint-type;
    description
          "Endpoint with cross-connect to an array
           of layer-3 adjacencies (variant:USP, USD and UET32).";
    }
        identity End.X_PSP_USP_USD_UET32 {
    base srv6-endpoint-type;
    description
          "Endpoint with cross-connect to an array
           of layer-3 adjacencies (variant:PSP, USP, USD and UET32).";
    }
        identity End.T_UET32 {
    base srv6-endpoin-type;
    description
          "Endpoint with specific IPv6 table lookup
          (variant: UET32).";
    }
        identity End.T_PSP_UET32 {
    base srv6-endpoin-type;
    description
          "Endpoint with specific IPv6 table lookup
          (variant: PSP and UET32).";
    }
        identity End.T_USP_UET32 {
    base srv6-endpoin-type;
    description
          "Endpoint with specific IPv6 table lookup
          (variant: USP and UET32).";
    }
        identity End.T_PSP_UET_UET32 {
    base srv6-endpoin-type;
    description
          "Endpoint with specific IPv6 table lookup
          (variant: PSP ,UET and UET32).";
    }
        identity End.T_USD_UET32 {
    base srv6-endpoint-type;
    description
```

```
          "Endpoint with specific IPv6 table lookup (variant:USD and UET32).";
    }
        identity End.T_PSP_USD_UET32 {
    base srv6-endpoint-type;
    description
          "Endpoint with specific IPv6 table lookup (variant:PSP, USD and
UET32).";
    }
        identity End.T_USP_USD_UET32 {
    base srv6-endpoint-type;
    description
          "Endpoint with specific IPv6 table lookup (variant:USP, USD and
UET32).";
    }
        identity End.T_PSP_USP_USD_UET32 {
    base srv6-endpoint-type;
    description
          "Endpoint with specific IPv6 table lookup(variant:PSP, USP, USD and
UET32).";
    }

        typedef uec-support-type {
    type enumeration {
     enum 128bit SID { value 1; description "The node only support to use
classical 128-bits SRv6 SID."; }
     enum 128-bits SRv6 SID and 32-bits IP U-SID { description "The node
support to use both classical 128-bits SRv6 SID and 32-bits IP U-SID"; }
     enum 128-bits SRv6 SID and 16-bits U-SID { description "The node support
to use both classical 128-bits SRv6 SID and 16-bits U-SID"; }
     enum 128-bits SRv6 SID, 32-bits IP U-SID, and 16-bits U-SID { description
"The node support to use both classical 128-bits SRv6 SID, 32-bits IP U-SID,
and 16-bits U-SID"; }
    }
    description
      "The node capabilities.";
  }

         typedef srv6-lb-length {
      type uint8 {
        range "8 .. 88";
      }
      description
         "This type defines an SRv6 locator block len with range constraints";
  }
        <CODE ENDS>
```

4.2. Shorter SRv6 Base

```
<CODEBEGINS> file "ietf-shorter-srv6-base@2020-03-30.yang"

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

   import ietf-srv6-base  {
```

```
    prefix "srv6";
    reference "RFC 8343: A YANG Data Model for SRv6 base.";
    }

    import ietf-srv6-types {
    prefix srv6-types;
    reference "RFC XXXX: YANG Data Model for SRv6 type";
    }
 import ietf-shhorter-srv6-types {
    prefix shourter-srv6-types;
    reference "RFC XXXX: YANG Data Model for shorter SRv6 type";
    }
    import ietf-routing {
    prefix "rt";
    reference
      "RFC 8349: A YANG Data Model for Routing Management
      (NMDA version)";
    }
    organization
       "IETF SPRING - SPRING Working Group";
     contact
       "WG Web:   <http://tools.ietf.org/wg/spring/>
        WG List:  <mailto:spring@ietf.org>
        Editor: Ran Chen
      <mailto:chen.ran@zte.com.cn>
     Editor: Detao Zhao
      <mailto:zhao.detao@zte.com.cn>
   ";
 description
         "The YANG module defines a generic configuration model for SFF.
    Copyright (c) 2019 IETF Trust and the persons
          identified as authors of the code.  All rights reserved.

          Redistribution and use in source and binary forms, with or
          without modification, is permitted pursuant to, and subject
          to the license terms contained in, the Simplified BSD License
          set forth in Section 4.c of the IETF Trust's Legal Provisions
          Relating to IETF Documents
          (https://trustee.ietf.org/license-info).
          This version of this YANG module is part of RFC XXXX; see
          the RFC itself for full legal notices.";

       revision "2020-03-30"{
      description "Initial revision.";
         reference "RFC XXXX: YANG Data Model for Shorter SRv6 .";
     }
    augment "/rt:routing/rt:srv6" {
   description
```

```
        "This augments SRv6 base data model with Shorter SRv6.";
  grouping shorter-srv6{
      description
              "SRv6 node capabilities grouping";
   container node-capabilities {
     description "SRv6 node capabilities."
               leaf uec-support-type{
       type shorter-srv6-types:uec-support-type;
    description "The node capabilities for UET."
                 }
    }
   }
  }
    augment "/rt:routing/rt:srv6/rt:locators/rt:locator/rt:prefix" {
    description
        "This augments SRv6 base data model with Shorter SRv6.";
  leaf srv6-lb-length{
     type shorter-srv6-types:srv6-lb-len;
       description "SRv6 locator block len with range constraints";
     }
    }
  }
  <CODE ENDS>
```

4.3. Shorter SRv6 Static

```
        <CODEBEGINS> file "ietf-shorter-srv6-type@2020-03-30.yang"
        module ietf-shorter-srv6-base {
        yang-version 1.1;
        namespace "urn:ietf:params:xml:ns:yang:ietf-shorter-srv6-base";
    prefix shorter-srv6;

    import ietf-srv6-base  {
    prefix "srv6";
    reference "RFC 8343: A YANG Data Model for SRv6 base.";
    }

    import ietf-srv6-types {
    prefix srv6-types;
    reference "RFC XXXX: YANG Data Model for SRv6";
    }
    import ietf-routing {
    prefix "rt";
    reference
      "RFC 8349: A YANG Data Model for Routing Management
      (NMDA version)";
    }
        import ietf-srv6-static  {
```

```
    prefix srv6-static ;
    reference "RFC XXXX: YANG Data Model for SRv6 static application.";
    }
        import ietf-shhorter-srv6-types {
    prefix shourter-srv6-types;
    reference "RFC XXXX: YANG Data Model for shorter SRv6 type";
    }
    organization
       "IETF SPRING - SPRING Working Group";
     contact
       "WG Web:   <http://tools.ietf.org/wg/spring/>
        WG List:  <mailto:spring@ietf.org>
        Editor: Ran Chen
                    <mailto:chen.ran@zte.com.cn>
            Editor: Detao Zhao
                    <mailto:zhao.detao@zte.com.cn>
                        ";
        description
         "The YANG module defines a generic configuration model for SFF.
                  Copyright (c) 2019 IETF Trust and the persons
          identified as authors of the code.  All rights reserved.

          Redistribution and use in source and binary forms, with or
          without modification, is permitted pursuant to, and subject
          to the license terms contained in, the Simplified BSD License
          set forth in Section 4.c of the IETF Trust's Legal Provisions
          Relating to IETF Documents
          (https://trustee.ietf.org/license-info).
          This version of this YANG module is part of RFC XXXX; see
          the RFC itself for full legal notices.";

       revision "2020-03-30"{
             description "Initial revision.";
         reference "RFC XXXX: YANG Data Model for Shorter SRv6 .";
     }


  grouping shorter-srv6-sid-config {
    description
      "Configuration parameters relating to Shorter SRv6 sid.";

    leaf function {
      type srv6-types:srv6-func-value;
      description
        "SRv6 function value.";
    }
    leaf end-behavior-type {
      type identityref {
```

```
        base shorter-srv6-types:srv6-endpoint-type;
     }
      mandatory true;
      description
        "Type of SRv6 end behavior.";
    }

        container end_uet32 {
      when "../end-behavior-type = 'End_UET32'" {
        description
          "This container is valid only when the user chooses End
           behavior (variant: UET32).";
      }
      description
        "The Endpoint function is the most basic function.
         FIB lookup on updated DA and forward accordingly
         to the matched entry.
         This is the SRv6 instantiation of a Prefix SID
         (variant: UET32)";
      }

         container end_psp_uet32 {
      when "../end-behavior-type = 'End_PSP_UET32'" {
        description
          "This container is valid only when the user chooses End
           behavior (variant: PSP and UET32).";
      }
      description
        "The Endpoint function is the most basic function.
         FIB lookup on updated DA and forward accordingly
         to the matched entry.
         This is the SRv6 instantiation of a Prefix SID
         (variant: PSP and UET32)";

    }
     container end_usp_uet32 {
      when "../end-behavior-type = 'End_USP_UET32'" {
        description
          "This container is valid only when the user chooses End
           behavior (variant: USP and UET32).";
      }
      description
        " This is the SRv6 instantiation of a Prefix SID
         (variant: USP and UET32)";

     container end_psp_usp_uet32 {
      when "../end-behavior-type = 'End_PSP_USP_UET32'" {
        description
```

```
          "This container is valid only when the user chooses End
           behavior (variant: PSP/USP/UET32).";
      }
      description
        " This is the SRv6 instantiation of a Prefix SID
         (variant: PSP/USP/UET32)";
    }
        }
        container end_usd_uet32 {
      when "../end-behavior-type = 'End_USD_UET32'" {
        description
          "This container is valid only when the user chooses End
           behavior (variant: USD/UET32).";
      }
      description
        "This is the SRv6 instantiation of a Prefix SID
         (variant: USD/UET32)";
    }

        container end_psp_usd_uet32 {
      when "../end-behavior-type = 'End_PSP_USD_UET32'" {
        description
          "This container is valid only when the user chooses End
           behavior (variant: PSP/USD/UET32).";
      }
      description
        "This is the SRv6 instantiation of a Prefix SID
         (variant: PSP/USD/UET32)";
      }
    container end_psp_usp_usd_uet32 {
      when "../end-behavior-type = 'End_PSP_USP_IUSD_UET32'" {
        description
          "This container is valid only when the user chooses End
           behavior (variant: PSP/USP/USD/UET32).";
      }
      description
        "This is the SRv6 instantiation of a Prefix SID
         (variant: PSP/USP/USD/UET32)";
    }

          container end-t_psp_uet32 {
      when "../end-behavior-type = 'End.T_PSP_UET32'" {
        description
          "This container is valid only when the user chooses
           End.T behavior (variant: PSP/UET32).";
      }
      description
        "Endpoint with specific IPv6 table lookup (variant: PSP/UET32).";
```

```
        leaf lookup-table-ipv6 {
          type srv6-types:table-id;
          mandatory true;
          description
            "Table Id for lookup on updated DA (next segment)";
        }
    }

         container end-t_uet32 {
      when "../end-behavior-type = 'End.T_UET32'" {
        description
          "This container is valid only when the user chooses
           End.T behavior (variant: UET32).";
      }
       description
        "Endpoint with specific IPv6 table lookup (variant: UET32).";
        leaf lookup-table-ipv6 {
          type srv6-types:table-id;
          mandatory true;
          description
            "Table Id for lookup on updated DA (next segment)";
        }
    }

  container end-t_usp_uet32 {
      when "../end-behavior-type = 'End.T_USP_UET32'" {
        description
          "This container is valid only when the user chooses
           End.T behavior (variant: USP/UET32).";
      }
      description
        "Endpoint with specific IPv6 table lookup (variant: USP/UET32).";
        leaf lookup-table-ipv6 {
          type srv6-types:table-id;
          mandatory true;
          description
            "Table Id for lookup on updated DA (next segment)";
        }
    }

     container end-t_psp_usp_uet32 {
      when "../end-behavior-type = 'End.T_PSP_USP_UET32'" {
        description
          "This container is valid only when the user chooses
           End.T behavior (variant: PSP/USP/UET32).";
      }
      description
        "Endpoint with specific IPv6 table lookup (variant: PSP/USP/UET32).";
```

```
        leaf lookup-table-ipv6 {
          type srv6-types:table-id;
          mandatory true;
          description
            "Table Id for lookup on updated DA (next segment)";
        }
    }

   container end-t_usd_uet32 {
      when "../end-behavior-type = 'End.T_USD_UET32'" {
        description
          "This container is valid only when the user chooses
           End.T behavior (variant: USD/UET32).";
      }
      description
        "Endpoint with specific IPv6 table lookup (variant: USD/UET32).";
        leaf lookup-table-ipv6 {
          type srv6-types:table-id;
          mandatory true;
          description
            "Table Id for lookup on updated DA (next segment)";
        }
    }

        container end-t_psp_usd_uet32 {
      when "../end-behavior-type = 'End.T_PSP_USD_UET32'" {
        description
          "This container is valid only when the user chooses
           End.T behavior (variant: PSP/USD/UET32).";
      }
      description
        "Endpoint with specific IPv6 table lookup (variant: PSP/USD/UET32).";
        leaf lookup-table-ipv6 {
          type srv6-types:table-id;
          mandatory true;
          description
            "Table Id for lookup on updated DA (next segment)";
        }
    }

        container end-t_usp_usd_uet32 {
      when "../end-behavior-type = 'End.T_USP_USD_UET32'" {
        description
          "This container is valid only when the user chooses
           End.T behavior (variant: USP/USD/UET32).";
      }
      description
        "Endpoint with specific IPv6 table lookup (variant: USP/USD/UET32).";
```

```
        leaf lookup-table-ipv6 {
          type srv6-types:table-id;
          mandatory true;
          description
            "Table Id for lookup on updated DA (next segment)";
        }
    }

        container end-t_psp_usp_usd_uet32 {
      when "../end-behavior-type = 'End.T_PSP_USP_USD_UET32'" {
        description
          "This container is valid only when the user chooses
           End.T behavior (variant: PSP/USP/USD/UET32).";
      }
      description
        "Endpoint with specific IPv6 table lookup (variant: PSP/USP/USD/
UET32)";
        leaf lookup-table-ipv6 {
          type srv6-types:table-id;
          mandatory true;
          description
            "Table Id for lookup on updated DA (next segment)";
        }
    }

        container end-x_uet32 {
      when "../end-behavior-type = 'End.X_UET32'" {
        description
          "This container is valid only when the user chooses
           End.X behavior (variant: UET32)";
      }
      description
        "Endpoint with cross-connect to an array of layer-3 adjacencies
(variant: UET32).";
      leaf protected {
        type boolean;
        default false;
        description "Is Adj-SID protected?";
      }
      uses srv6-static:multi-paths-v6;
    }

        container end-x_psp_uet32 {
      when "../end-behavior-type = 'End.X_PSP_UET32'" {
        description
          "This container is valid only when the user chooses
           End.X behavior (variant: PSP/UET32)";
      }
```

```
      description
        "Endpoint with cross-connect to an array of layer-3 adjacencies
(variant: PSP/UET32).";
```

```
      leaf protected {
        type boolean;
        default false;
        description "Is Adj-SID protected?";
      }

      uses srv6-static:multi-paths-v6;
    }

    container end-x_usp_uet32 {
      when "../end-behavior-type = 'End.X_USP_UET32'" {
        description
          "This container is valid only when the user chooses
           End.X behavior (variant: USP/UET32)";
      }
      description
        "Endpoint with cross-connect to an array of layer-3 adjacencies
(variant: USP/UET32).";

      leaf protected {
        type boolean;
        default false;
        description "Is Adj-SID protected?";
      }

      uses srv6-static:multi-paths-v6;
    }

   container end-x_psp_usp_uet32 {
      when "../end-behavior-type = 'End.X_PSP_USP_UET32'" {
        description
          "This container is valid only when the user chooses
           End.X behavior (variant: PSP/USP/UET32)";
      }
      description
        "Endpoint with cross-connect to an array of layer-3 adjacencies
(variant: PSP/USP/UET32).";

      leaf protected {
        type boolean;
        default false;
        description "Is Adj-SID protected?";
      }

      uses srv6-static:multi-paths-v6;
    }

        container end-x_usd_uet32 {
      when "../end-behavior-type = 'End.X_USD_UET32'" {
```

description

```
          "This container is valid only when the user chooses
           End.X behavior (variant: USD/UET32)";
      }
      description
        "Endpoint with cross-connect to an array of layer-3 adjacencies
(variant: USD/UET32).";

      leaf protected {
        type boolean;
        default false;
        description "Is Adj-SID protected?";
      }
      uses srv6-static:multi-paths-v6;
    }
        container end-x_psp_usd_uet32 {
      when "../end-behavior-type = 'End.X_PSP_USD_UET32'" {
        description
          "This container is valid only when the user chooses
           End.X behavior (variant: PSP/USD/UET32)";
      }
      description
        "Endpoint with cross-connect to an array of layer-3 adjacencies
(variant: PSP/USD/UET32).";

      leaf protected {
        type boolean;
        default false;
        description "Is Adj-SID protected?";
      }

      uses srv6-static:multi-paths-v6;
    }
        container end-x_usp_usd_uet32 {
      when "../end-behavior-type = 'End.X_USP_USD_UET32'" {
        description
          "This container is valid only when the user chooses
           End.X behavior (variant: USP/USD/UET32)";
      }
      description
        "Endpoint with cross-connect to an array of layer-3 adjacencies
(variant: USP/USD/UET32).";

      leaf protected {
        type boolean;
        default false;
        description "Is Adj-SID protected?";
      }
```

```
      uses srv6-static:multi-paths-v6;
    }
```

```
         container end-x_psp_usp_usd_uet32 {
      when "../end-behavior-type = 'End.X_PSP_USP_USD_UET32'" {
        description
          "This container is valid only when the user chooses
           End.X behavior (variant: PSP/USP/USD/UET32)";
      }
      description
        "Endpoint with cross-connect to an array of layer-3 adjacencies
(variant: PSP/USP/USD/UET32).";

      leaf protected {
        type boolean;
        default false;
        description "Is Adj-SID protected?";
      }

      uses srv6-static:multi-paths-v6;
     }
        }

        grouping shorter-srv6-static-cfg {
    description
      "Grouping configuration and operation for Shorter SRv6 sid.";
    list sid {
       key "function";
       description "List of locally instantiated SIDs";
        uses shorter-srv6-sid-config;
      }
  }
        augment"/rt:routing/srv6:srv6/srv6:locators/srv6:locator:/srv6:static/
srv6:local-sids/srv6:sid"{
           description
          "This augments SRv6 static data model with Shorter static SRv6.";
     leaf end-behavior-type {
       type identityref {
         base shorter-srv6-types:srv6-endpoint-type;
          }
         mandatory true;
         description
           "Type of SRv6 end behavior.";
       }
            uses shorter-srv6-static-cfg;
     }
        }
                <CODE ENDS>
```

5. Security Considerations

TBD.

6. IANA Considerations

TBD.

7. Normative References

[I-D.ietf-6man-segment-routing-header]
Filsfils, C., Dukes, D., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-26 (work in progress), October 2019.

[I-D.ietf-spring-srv6-network-programming]
Filsfils, C., Camarillo, P., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-ietf-spring-srv6-network-programming-24 (work in progress), October 2020.

[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", draft-ietf-spring-srv6-yang-00 (work in progress), September 2020.

[I-D.mirsky-6man-unified-id-sr]
Cheng, W., Mirsky, G., Peng, S., Aihua, L., and G. Mishra, "Unified Identifier in IPv6 Segment Routing Networks", draft-mirsky-6man-unified-id-sr-07 (work in progress), July 2020.

[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>.

Authors' Addresses

Ran Chen
ZTE Corporation
No. 50 Software Ave, Yuhuatai Distinct
Nanjing, P.R.China
China

Email: chen.ran@zte.com.cn

Detao zhao
ZTE Corporation
No. 50 Software Ave, Yuhuatai Distinct
Nanjing, P.R.China
China

Email: zhao.detao@zte.com.cn