Spring
Internet-Draft
Intended status: Standards Track
Expires: 18 April 2024

M. Jethanandani, Ed.
T. Murakami
K. Rajaram
Arrcus, Inc
S. Matsushima
SoftBank

16 October 2023

A YANG Data Model for SRv6 Mobile User Plane
draft-mahesh-bess-srv6-mup-yang-03

Abstract

This document defines a YANG data model for configuration and management of Mobile User Plane (MUP) in a SRv6 network.

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 18 April 2024.

Copyright Notice

Table of Contents

1. Introduction

In mobile networks, mobility systems provide connectivity over a wireless link to stationary and non-stationary nodes. The user-plane establishes a tunnel between the mobile node and its anchor node over IP-based backhaul and core networks.

When SRv6 is applied to mobile networks, it enables a source routing architecture, where operators get to explicitly specify a route for the packets to traverse both to and from a mobile node. The SRv6 Endpoint nodes serve as mobile user-plane anchors.

Segment Routing IPv6 Mobile User Plane Architecture For Distributed Mobility Management [I-D.mhkk-dmm-srv6mup-architecture], defines the Segment Routing IPv6 Mobile User Plane (SRv6 MUP) architecture for Distributed Mobility Management. As part of the architecture, the document defines a new SRv6 segment type called as a MUP Segment, new routing information that can carried within BGP, and advertised from a PE and a MUP Controller. BGP Extensions for Mobile User Plane (MUP) SAFI [I-D.mpmz-bess-mup-safi] further defines a new Subsequent Address Family Indicator (SAFI) for the BGP Mobile User Plane (BGP-MUP) to support MUP extensions for BGP.

This document defines a YANG 1.1 [RFC7950] data model for BGP-MUP. The model conforms to the NMDA [RFC8342] architecture.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.2. Note to RFC Editors

This document uses several placeholder values throughout the document. Please replace them as follows and remove this note before publication.

RFC XXXX, where XXXX is the number assigned to this document at the time of publication.

2023-10-16 with the actual date of the publication of this document.

2. Terminology

This document references terms defined in other documents. In particular, it imports definitions for the following terms from Segment Routing Architecture [RFC8402].

* Active Segment
* BGP-Prefix Segment
* Segment
* SRv6
* Segment Routing domain (SR domain)

2.1. Acronyms

This document uses a few acronyms. Some of them are defined here for reference.

Acronym	Definition
MUP	Mobile User Plane
RD	Route Distinguisher
RT	Route Target
SAFI	Subsequent Address Family Indicator
SR	Segment Routing
SRv6	Segment Routing over v6
VRF	Virtual Routing and Forwarding

Table 1: Acronyms

3. Tree Diagram

An abridged version of the tree diagram is shown here. Annotations used in the diagram are defined in YANG Tree Diagrams [RFC8340].

```
module: ietf-mup

  augment /rt:routing/rt:control-plane-protocols
            /rt:control-plane-protocol/bgp:bgp/bgp:global
            /bgp:afi-safis/bgp:afi-safi:
    +--rw ipv4-mup
    |  +--rw rts
    |  |  +--rw rt* [name]
    |  |        ...
    |  +--rw routing-table-limit
    |  |  +--rw routes* [type]
    |  |        ...
    |  +--rw segment* [type]
    |  |  +--rw type            identityref
    |  |  +--rw locator?        leafref
    |  |  +--rw entry*          union
    |  |  +--rw mup-ext-comm*   mup-ext-community-type
    |  +--rw architecture-type?     identityref
    +--rw ipv6-mup
       +--rw rts
       |  +--rw rt* [name]
       |        ...
       +--rw routing-table-limit
```

```
          |  +--rw routes* [type]
          |        ...
          +--rw segment* [type]
          |  +--rw type            identityref
          |  +--rw locator?        leafref
          |  +--rw entry*          union
          |  +--rw mup-ext-comm*   mup-ext-community-type
          +--rw architecture-type?     identityref
     augment /rt:routing/rt:control-plane-protocols
               /rt:control-plane-protocol/bgp:bgp/bgp:global
               /bgp:afi-safis/bgp:afi-safi/bgp:ipv4-unicast:
       +--rw rts
          +--rw rt* [name]
             +--rw name             identityref
             +--rw route-policy?    leafref
             +--rw route-targets
                    ...
     augment /rt:routing/rt:control-plane-protocols
               /rt:control-plane-protocol/bgp:bgp/bgp:global
               /bgp:afi-safis/bgp:afi-safi/bgp:ipv6-unicast:
       +--rw rts
          +--rw rt* [name]
             +--rw name             identityref
             +--rw route-policy?    leafref
             +--rw route-targets
                    ...
     augment /rt:routing/rt:control-plane-protocols
               /rt:control-plane-protocol/bgp:bgp/bgp:global
               /bgp:afi-safis/bgp:afi-safi/bgp:l3vpn-ipv4-unicast:
       +--rw rts
          +--rw rt* [name]
             +--rw name             identityref
             +--rw route-policy?    leafref
             +--rw route-targets
                    ...
     augment /rt:routing/rt:control-plane-protocols
               /rt:control-plane-protocol/bgp:bgp/bgp:global
               /bgp:afi-safis/bgp:afi-safi/bgp:l3vpn-ipv6-unicast:
       +--rw rts
          +--rw rt* [name]
             +--rw name             identityref
             +--rw route-policy?    leafref
             +--rw route-targets
                    ...
     augment /rt:routing/rt:control-plane-protocols
               /rt:control-plane-protocol/bgp:bgp/bgp:neighbors
               /bgp:neighbor/bgp:afi-safis/bgp:afi-safi:
       +--rw ipv4-mup!
```

```
  +--rw ipv6-mup!
augment /rt:routing/rt:control-plane-protocols
          /rt:control-plane-protocol/bgp:bgp/bgp:neighbors
          /bgp:neighbor/bgp:statistics:
  +--ro isd-sent?             yang:zero-based-counter32
  +--ro isd-received?         yang:zero-based-counter32
  +--ro dsd-sent?             yang:zero-based-counter32
  +--ro dsd-received?         yang:zero-based-counter32
  +--ro type-1-st-sent?       yang:zero-based-counter32
  +--ro type-1-st-received?   yang:zero-based-counter32
  +--ro type-2-st-sent?       yang:zero-based-counter32
  +--ro type-2-st-received?   yang:zero-based-counter32
augment /rt:routing/rt:control-plane-protocols
          /rt:control-plane-protocol/bgp:bgp/bgp:global:
  +--rw srv6
     +--rw locator?                 leafref
     +--ro sid-manager-connected?   boolean
     +--ro locator-registered?      boolean
     +--ro micro-segment-enabled?   boolean
     +--rw sid-allocation-mode?     identityref
     +--rw mobile
        +--rw encapsulation
        |     ...
        +--rw decapsulations
              ...
augment /ni:network-instances/ni:network-instance/ni:ni-type:
  +--:(mup)
     +--rw mup
        +--rw rd
              ...
augment /rt-pol:routing-policy/rt-pol:policy-definitions
          /rt-pol:policy-definition/rt-pol:statements
          /rt-pol:statement/rt-pol:conditions/bp:bgp-conditions:
  +--rw match-mup
     +--rw route-type?   identityref
augment /rt-pol:routing-policy/rt-pol:policy-definitions
          /rt-pol:policy-definition/rt-pol:statements
          /rt-pol:statement/rt-pol:actions:
  +--rw mup-actions
     +--rw accept-route?   boolean
```

Figure 1: Tree Diagram for SRv6 YANG Model

4. YANG Model

The YANG model is divided into two parts. The first and the main part of the model augments the BGP model in YANG Model for Border Gateway Protocol (BGP-4) [I-D.ietf-idr-bgp-model] for the BGP configuration, while the second part augments the Network Instance YANG model in YANG Data Model for Network Instances [RFC8529] to add in configuration related to MUP at a VRF level, e.g., Route Distinguisher (RD).

This model defines two new SAFIs called 'ipv4-mup' and 'ipv6-mup'. The BGP model is augmented at a global, and at a neighbor level to add MUP configuration. In addition, containers for AFI/SAFI of type 'ipv4-unicast', 'ipv6-unicast', 'l3vpn-ipv4-unicast', and 'l3vpn-ipv6-unicast' are augmented to add Route Targets (RT). Finally, the Network Instance model is augmented to add VRF specific configuration for the MUP segments.

The model imports Common YANG Data Types [RFC6991], Common YANG Data Types for the Routing Area [RFC8294], A YANG Data Model for Interface Management [RFC8343], A YANG Data Model for Routing Management(NMDA Version) [RFC8349], YANG Data Model for Network Instances [RFC8529], A YANG Data Model for Routing Policy [RFC9067], YANG Data Model for Segment Routing [RFC9020], A YANG Data Model for Routing Policy [RFC9067], YANG Data Model for SRv6 Base and Static [I-D.ietf-spring-srv6-yang], and BGP Model for Service Provider Network [I-D.ietf-idr-bgp-model].

```
<CODE BEGINS> file "ietf-mup@2023-10-16.yang"
module ietf-mup {
  yang-version "1.1";
  namespace "urn:ietf:params:xml:ns:yang:ietf-mup";
  prefix "ietf-mup";

  import ietf-yang-types {
    prefix yang;
    reference
      "RFC 6991: Common YANG Data Types.";
  }
  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 6991: Common YANG Data Types.";
  }
  import ietf-interfaces {
    prefix if;
    reference
      "RFC 8343: YANG Data Model for Interface Management.";
```

```
  }
  import ietf-routing {
    prefix rt;
    reference
      "RFC 8349, A YANG Data Model for Routing Management
       (NMDA Version).";
  }
  import ietf-routing-types {
    prefix rt-types;
    reference
      "RFC 8294: Common YANG Data Types for the Routing Area.";
  }
  import ietf-routing-policy {
    prefix rt-pol;
    reference
      "RFC 9067: A YANG Data Model for Routing Policy.";
  }
  import ietf-network-instance {
    prefix ni;
    reference
      "RFC 8529: YANG Data Model for Network Instance.";
  }
  import ietf-bgp {
    prefix bgp;
    reference
      "I-D.ietf-idr-bgp-model: YANG Data Model for Border
                               Gateway Protocol 4 (BGP-4).";
  }
  import iana-bgp-types {
    prefix bt;
    reference
      "I-D.ietf-idr-bgp-model:  YANG Data Model for Border
                                Gateway Protocol 4 (BGP-4).";
  }
  import ietf-bgp-policy {
    prefix bp;
    reference
      "I-D.ietf-idr-bgp-model:  YANG Data Model for Border
                                Gateway Protocol 4 (BGP-4).";
  }
  import ietf-segment-routing {
    prefix sr;
    reference
      "RFC 9020: YANG Data Model for Segment Routing.";
  }
  import ietf-srv6-base {
    prefix srv6;
    reference
```

```
       "I-D.ietf-spring-srv6-yang: YANG Data Model for SRv6 Base
                                   and Static.";
   }

   organization
     "IETF BESS Working Group";

   contact
     "WG Web:   <https://datatracker.ietf.org/wg/bess/about>
      WG List:  <bess@ietf.org>

      Editor: Mahesh Jethanandani (mjethanandani at gmail dot com)
      Author:  Tetsuya Murakami (tetsuya at arrcus dot com)";

   description
     "This module augments the BGP YANG model to add support for
      configuration in mobile networks.

      Copyright (c) 2022 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 Revised 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
      (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
      for full legal notices.

      The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL
      NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'NOT
      RECOMMENDED', 'MAY', and 'OPTIONAL' in this document are to
      be interpreted as described in BCP 14 (RFC 2119) (RFC 8174)
      when, and only when, they appear in all capitals, as shown
      here.";

   revision "2023-10-16" {
     description
       "Initial Version.";
     reference
       "RFC XXXX, A YANG Model for BGP configuration in mobile
        networks.";
   }

  /*
```

```
   * Identities
   */
   identity sid-allocation-mode {
     description
       "Base identity to be used to express types of SRv6 segment ID
        allocation strategies.";
   }

   identity sid-per-nexthop {
     base sid-allocation-mode;
     description
       "A segment ID is allocated per nexthop entry in the RIB.";
   }

   identity instance-sid {
     base sid-allocation-mode;
     description
       "A single segment ID is used.";
   }

   /*
    * Typedefs
    */
   typedef mup-ext-community-type {
     type string {
       // (ASN):(local-part)
       pattern '(6553[0-5]|655[0-2][0-9]|654[0-9]{2}|65[0-4][0-9]{2}' +
               '|6[0-4][0-9]{3}|[1-5][0-9]{4}|[1-9][0-9]{1,3}|[0-9]):' +
               '(429496729[0-5]|42949672[0-8][0-9]|4294967[0-1][0-9]' +
               '{2}|429496[0-6][0-9]{3}|42949[0-5][0-9]{4}|4294[0-8]' +
               '[0-9]{5}|429[0-3][0-9]{6}|4[0-1][0-9]{7}|[1-3][0-9]' +
               '{9}|[1-9][0-9]{1,8}|[0-9])';
     }
     description
       "A type definition utilised to define the extended community
        in routes of Mobile User Plane (MUP) SAFI. The above pattern
        is a placeholder regex, till such time that a format for it has
        been defined.";
   }

   /*
    * Identities
    */
   identity ipv4-mup {
     base bt:afi-safi-type;
     description
       "AFI/SAFI for Mobile User Plane (AFI,SAFI = 1, 85)";
     reference
```

```
       "RFC XXXX: A YANG Model for BGP configuration of Mobile
                  User Plane (MUP).";
   }

   identity ipv6-mup {
     base bt:afi-safi-type;
     description
       "AFI/SAFI for Mobile User Plane (AFI,SAFI = 2, 85)";
     reference
       "RFC XXXX: A YANG Model for BGP configuration of Mobile
                  User Plane (MUP).";
   }

   identity architecture-type {
     description
       "Base identity for Architecture Type.";
     reference
       "I-D.mpmz-bess-mup-safi: BGP Extensions for the Mobile
                                User Plane (MUP) SAFI.";
   }

   identity three-gpp-5g {
     base architecture-type;
     description
       "The Architecture Type for BGP-MUP NLRI.";
     reference
       "I-D.mpmz-bess-mup-safi: BGP Extensions for the Mobile
                                User Plane (MUP) SAFI.";
   }

   identity route-type {
     description
       "Base identity for Route Type.";
     reference
       "I-D.mpmz-bess-mup-safi: BGP Extensions for the Mobile
                                User Plane (MUP) SAFI.";
   }

   identity segment-type {
     description
       "Base identity for Segment Type.";
    reference
       "I-D.mpmz-bess-mup-safi: BGP Extensions for the Mobile
                                User Plane (MUP) SAFI.";
   }

   identity isd {
     base route-type;
```

```
    base segment-type;
    description
      "The Interwork Segment Discovery Route Type.";
    reference
      "I-D.mpmz-bess-mup-safi: BGP Extensions for the Mobile
                               User Plane (MUP) SAFI.";
  }

  identity dsd {
    base route-type;
    base segment-type;
    description
      "The Direct Segment Discovery Route Type.";
    reference
      "I-D.mpmz-bess-mup-safi: BGP Extensions for the Mobile
                               User Plane (MUP) SAFI.";
  }

  identity type-1-st {
    base route-type;
    description
      "Type 1 Session Transformed (ST) Route Type.";
    reference
      "I-D.mpmz-bess-mup-safi: BGP Extensions for the Mobile
                               User Plane (MUP) SAFI.";
  }

  identity type-2-st {
    base route-type;
    description
      "Type 2 Session Transformed (ST) Route Type.";
    reference
      "I-D.mpmz-bess-mup-safi: BGP Extensions for the Mobile
                               User Plane (MUP) SAFI.";
  }

  /*
   * Groupings
   */
  grouping rts {
    description
      "Grouping for configuration of route targets for AFI/SAFIs.";

    container rts {
      description
        "Container for configuration of Route Targets for address
         family list.";
```

```
       list rt {
         key "name";
         description
           "List of route targets for a given afi-safi type.";

         leaf name {
           type identityref {
             base bt:afi-safi-type;
           }
           must "derived-from-or-self(., 'ipv4-mup')  or " +
                "derived-from-or-self(., 'ipv6-mup')" {
             error-message
               "Only ipv4-mup or ipv6-mup are supported.";
           }
           description
             "Name of the AFI/SAFI type.";
         }

         leaf route-policy {
           type leafref {
             path "/rt-pol:routing-policy" +
                  "/rt-pol:policy-definitions/" +
                  "rt-pol:policy-definition/rt-pol:name";
             require-instance true;
           }
           description
             "Reference to the route policy containing set of
              route-targets.";
         }

         container route-targets {
           description
             "Route Targets for a network instance.";

           list route-target {
             key "target type";

             description
               "List of route targets.";

             leaf target {
               type rt-types:route-target;
               description
                 "A Route Target is an 8-octet BGP extended community
                  initially identifying a set of sites in a BGP VPN
                  (RFC 4364).  However, it has since taken on a more
                  general role in BGP route filtering.  A Route Target
                  consists of two or three fields: a 2-octet Type
```

```
                  field, an administrator field, and, optionally, an
                  assigned number field.

                  According to the data formats for types 0, 1, 2, and
                  6 as defined in RFC 4360, RFC 5668, and RFC 7432,
                  the encoding pattern is defined as:

                  0:2-octet-asn:4-octet-number
                  1:4-octet-ipv4addr:2-octet-number
                  2:4-octet-asn:2-octet-number
                  6:6-octet-mac-address

                  Additionally, a generic pattern is defined for future
                  Route Target types:

                  2-octet-other-hex-number:6-octet-hex-number

                  Some valid examples are 0:100:100, 1:1.1.1.1:100,
                  2:1234567890:203, and 6:26:00:08:92:78:00.";
              }

              leaf type {
                type rt-types:route-target-type;
                description
                  "Reference to route-target type.";
              }

            }
          }
        }
      }
    }

    grouping bgp-mup {
      description
        "BGP-MUP NLRI configuration.";

      uses rts;

      container routing-table-limit {
        description
          "The routing-table limit command sets a limit on the maximum
           number of routes imported that the IPv4 or IPv6 address
           family of a MUP instance can support.

           By default, there is no limit on the maximum number of
           routes that the IPv4 or IPv6 address family of a MUP
           instance can support, but the total number of private
```

```
          network and public network routes on a device cannot
          exceed the allowed maximum number of unicast routes.";

        list routes {
          key "type";
          description
            "List of routes that need to be limited by type.";

          leaf type {
            type identityref {
              base route-type;
            }
            description
              "The type of route on which a limit is being placed.";
          }

          leaf number {
            type uint32 {
              range "1..max";
            }
            description
              "Specifies the maximum number of routes supported by a
               MUP instance. ";
          }

          choice action {
            description
              "Choice of actions to take.";

            leaf percent {
              type rt-types:percentage;
              description
                "Specifies the percentage of the maximum number of
                 routes. When the maximum number of routes that join
                 the MUP instance is up to the value
                 (number*percent)/100, the system prompts alarms.
                 The MUP routes can be still added to the routing
                 table, but after the number of routes reaches number,
                 the subsequent routes are dropped.";
            }

            leaf simple {
              type boolean;
              description
                "Indicates that when MUP routes exceed number, routes
                 can still be added into the routing table, but the
                 system prompts alarms.
```

```
                However, after the total number of VPN routes and
                network public routes reaches the unicast route limit
                specified in the License, the subsequent routes
                are dropped.";
            }
          }
        }
      }

      list segment {
        key "type";
        description
          "List of segments.";

        leaf type {
          type identityref {
            base segment-type;
          }
          description
            "Type of segment.";
        }

        leaf locator {
          type leafref {
            path "/rt:routing/sr:segment-routing/srv6:srv6" +
                 "/srv6:locators/srv6:locator/srv6:name";
          }
          must "derived-from-or-self(../type, 'isd')";
          description
            "Reference to locator in the 'default' VRF
             configuration.";
        }

        leaf-list entry {
          type union {
            type inet:ip-prefix;
            type if:interface-ref;
            type enumeration {
              enum router-ip {
                description
                  "Entry is of type router-ip.";
              }
            }
          }
          description
            "MUP entries.";
        }
```

```
      leaf-list mup-ext-comm {
        type mup-ext-community-type;
        must "derived-from-or-self(../type, 'dsd')";
        description
          "MUP extended community type.";
      }
    }

    leaf architecture-type {
      type identityref {
        base architecture-type;
      }
      description
        "Encoding of the rest of BGP-MUP NLRI for a given
         MUP architecture.";
    }
  }

  /*
   * BGP configuration
   */
  augment "/rt:routing/rt:control-plane-protocols" +
          "/rt:control-plane-protocol/bgp:bgp/bgp:global" +
          "/bgp:afi-safis/bgp:afi-safi" {
    description
      "Augmentation of the BGP model to add BGP-MUP.";

    container ipv4-mup {
      when "derived-from-or-self (../../bgp:afi-safi/bgp:name,
           'ipv4-mup')" {
        description
          "This configuration applies only if the identity is
           IPv4 MUP.";
      }

      uses bgp-mup;
      description
        "IPv4 MUP configuration and management.";
    }

    container ipv6-mup {
      when "derived-from-or-self (../../bgp:afi-safi/bgp:name,
           'ipv6-mup')" {
        description
          "This configuration applies only if the identity is
           IPv6 MUP.";
      }
```

```
      uses bgp-mup;
      description
        "IPv6 MUP configuration and management.";
    }
  }

  augment "/rt:routing/rt:control-plane-protocols" +
          "/rt:control-plane-protocol/bgp:bgp/bgp:global" +
          "/bgp:afi-safis/bgp:afi-safi/bgp:ipv4-unicast" {
    description
      "Augmentation of the BGP model to add RT for ipv4-unicast.";

    uses rts;
  }

  augment "/rt:routing/rt:control-plane-protocols" +
          "/rt:control-plane-protocol/bgp:bgp/bgp:global" +
          "/bgp:afi-safis/bgp:afi-safi/bgp:ipv6-unicast" {
    description
      "Augmentation of the BGP model to add RT for ipv6-unicast.";

    uses rts;
  }

  augment "/rt:routing/rt:control-plane-protocols" +
          "/rt:control-plane-protocol/bgp:bgp/bgp:global" +
          "/bgp:afi-safis/bgp:afi-safi/bgp:l3vpn-ipv4-unicast" {
    description
      "Augmentation of the BGP model to add RT for
       l3vpn-ipv4-unicast.";

    uses rts;
  }

  augment "/rt:routing/rt:control-plane-protocols" +
          "/rt:control-plane-protocol/bgp:bgp/bgp:global" +
          "/bgp:afi-safis/bgp:afi-safi/bgp:l3vpn-ipv6-unicast" {
    description
      "Augmentation of the BGP model to add RT for
       l3vpn-ipv6-unicast.";

    uses rts;
  }

  augment "/rt:routing/rt:control-plane-protocols" +
          "/rt:control-plane-protocol/bgp:bgp/bgp:neighbors" +
          "/bgp:neighbor/bgp:afi-safis/bgp:afi-safi" {
    description
```

```
      "Augmentation of the BGP model to add BGP-MUP.";

    container ipv4-mup {
      when "derived-from-or-self(../../bgp:afi-safi/bgp:name,
            'ipv4-mup')" {
        description
          "This configuration applies only if the identity is
           IPv4 MUP.";
      }
      presence "Presence container for IPv4 MUP.";
      description
        "IPv4 MUP configuration and management on a per neighbor
         basis.";
    }

    container ipv6-mup {
      when "derived-from-or-self(../../bgp:afi-safi/bgp:name,
            'ipv6-mup')" {
        description
          "This configuration applies only if the identity is
           IPv6 MUP.";
      }
      presence "Presence container for IPv6 MUP.";
      description
        "IPv6 MUP configuration and management on a per neighbor
         basis.";
    }
  }

  augment "/rt:routing/rt:control-plane-protocols" +
          "/rt:control-plane-protocol/bgp:bgp/bgp:neighbors" +
          "/bgp:neighbor/bgp:statistics" {
    description
      "Augmentation of the BGP per-neighbor statistics to add
       BGP-MUP specific counters.";

    leaf isd-sent {
      type yang:zero-based-counter32;
      description
        "Total number of BGP Interwork Segment Discovery routes sent
         per neighbor.";
    }

    leaf isd-received {
      type yang:zero-based-counter32;
      description
        "Total number of BGP Interwork Segment Discovery routes
         received per neighbor.";
```

```
    }

    leaf dsd-sent {
      type yang:zero-based-counter32;
      description
        "Total number of BGP Direct Segment Discovery routes sent
         per neighbor.";
    }

    leaf dsd-received {
      type yang:zero-based-counter32;
      description
        "Total number of BGP Direct Segment Discovery routes
         received per neighbor.";
    }

    leaf type-1-st-sent {
      type yang:zero-based-counter32;
      description
        "Total number of BGP Type 1 Session Transformed routes sent
         per neighbor.";
    }

    leaf type-1-st-received {
      type yang:zero-based-counter32;
      description
        "Total number of BGP Type 1 Session Transformed routes
         received per neighbor.";
    }

    leaf type-2-st-sent {
      type yang:zero-based-counter32;
      description
        "Total number of BGP Type 2 Session Transformed routes sent
         per neighbor.";
    }

    leaf type-2-st-received {
      type yang:zero-based-counter32;
      description
        "Total number of BGP Type 2 Session Transformed routes
         received per neighbor.";
    }
  }

  augment "/rt:routing/rt:control-plane-protocols" +
          "/rt:control-plane-protocol/bgp:bgp/bgp:global" {
    description
```

```
       "Augmentation of the BGP model to add SRv6 mobile
        configuration.";

     container srv6 {
       description
         "Container to define SRv6 MUP configuration.";

       leaf locator {
         type leafref {
           path "/rt:routing/sr:segment-routing/srv6:srv6" +
                "/srv6:locators/srv6:locator/srv6:name";
         }
         description
          "Reference to the locator configured.";
       }

       leaf sid-manager-connected {
         type boolean;
         config false;
         description
           "Is the connection with segment ID manager active?";
       }

       leaf locator-registered {
         type boolean;
         config false;
         description
           "Is the locator name registered?";
       }

       leaf micro-segment-enabled {
         type boolean;
         config false;
         description
           "Is the locator enabled for micro-segment behavior?";
       }

       leaf sid-allocation-mode {
         type identityref {
           base sid-allocation-mode;
         }
         must "boolean((../mobile/encapsulation/locator) or " +
              "(../mobile/decapsulations/decapsulation/locator) or " +
              "(../locator))" {
           error-message
             "SRv6 locator name must be configured";
         }
         description
```

```
            "The segment ID allocation mode to be used for L3 entries
             in the network instance";
          }

          container mobile {
            description
              "MUP configuration.";

            container encapsulation {
              description
                "Encapsulation configuration for mobile data.";

              leaf locator {
                type leafref {
                  path "../../../locator";
                }
                description
                  "Encapsulation specific locator.";
              }

            }

            container decapsulations {
              description
                "Container for all SRv6 mobile decapsulation
                 configurations";

              list decapsulation {
                key "id";

                description
                  "SRv6 mobile decapsulation config";

                leaf id {
                  type uint16 {
                    range "1..100";
                  }
                  description
                    "SRv6 mobile decapsulation entry id";
                }

                leaf locator {
                  type leafref {
                    path "../../../../locator";
                  }
                  description
                    "Encapsulation specific locator.";
                }
```

```
            container mup-ext-communities {
              description
                "MUP extended communities configuration.";

              leaf-list mup-ext-community {
                type mup-ext-community-type;
                description
                  "List of MUP extended communities.";
              }
            }
          }
        }
      }
    }

    /*
     * Network Instance configuration.
     */
    augment "/ni:network-instances/ni:network-instance/ni:ni-type" {
       description
         "Augment network instance for per VRF MUP parameters";

       case mup {
        container mup {
          description
            "Configuration of MUP specific parameters";

          container rd {
            description
              "Route distinguisher parameters.";

            leaf rd {
              type union {
                type rt-types:route-distinguisher;
                type enumeration {
                  enum auto {
                    description
                      "Route distinguisher is assigned automatically.";
                  }
                }
              }
              description
                "Route distinguisher value.";
              reference
                "RFC 4364: BGP/MPLS IP Virtual Private Networks
                           (VPNs).";
            }
```

```
          leaf auto-rd {
            type rt-types:route-distinguisher;
            config false;
            description
              "Automatically assigned RD value when rd is configured
               as 'auto'.";
          }
        }
      }
    }
  }

  augment "/rt-pol:routing-policy/rt-pol:policy-definitions/" +
          "rt-pol:policy-definition/rt-pol:statements/" +
          "rt-pol:statement/rt-pol:conditions/bp:bgp-conditions" {
    description
      "BGP policy conditions added to routing policy module.";

    container match-mup {
      description
        "Top-level container for MUP specific policy conditions.";

      leaf route-type {
        type identityref {
          base route-type;
        }
        description
          "Route type to match with.";
      }
    }
  }

  augment "/rt-pol:routing-policy/rt-pol:policy-definitions/" +
          "rt-pol:policy-definition/rt-pol:statements/" +
          "rt-pol:statement/rt-pol:actions" {
    description
      "MUP policy actions added to routing policy module.";

    container mup-actions {
      description
        "Container for adding MUP specific actions.";

      leaf accept-route {
        type boolean;
        default false;
        description
          "Accept this route.";
      }
```

```
     }
   }
 }
 <CODE ENDS>
```

Figure 2: YANG Model for Mobile User Plane

5. IANA Considerations

This memo registers the following namespace URIs in the IETF XML in the "IETF XML Registry" [RFC3688]:

URI: urn:ietf:params:xml:ns:yang:ietf-mup
Registrant Contact: The IESG.
XML: N/A; the requested URI is an XML namespace.

This document registers the following YANG modules in the "YANG Module Names" registry [RFC6020]:

Name: ietf-mup
Namespace: urn:ietf:params:xml:ns:yang:ietf-mup
Prefix: srv6-mob
Reference: RFC XXXX

6. Security Considerations

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

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

Some of the RPC operations in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations. These are the operations and their sensitivity/vulnerability:

7. References

7.1. Normative References

[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>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <https://www.rfc-editor.org/info/rfc6242>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <https://www.rfc-editor.org/info/rfc6991>.

[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>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

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

[RFC8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Common YANG Data Types for the Routing Area", RFC 8294, DOI 10.17487/RFC8294, December 2017, <https://www.rfc-editor.org/info/rfc8294>.

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

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <https://www.rfc-editor.org/info/rfc8341>.

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

[RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <https://www.rfc-editor.org/info/rfc8343>.

[RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <https://www.rfc-editor.org/info/rfc8349>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8529] Berger, L., Hopps, C., Lindem, A., Bogdanovic, D., and X. Liu, "YANG Data Model for Network Instances", RFC 8529, DOI 10.17487/RFC8529, March 2019, <https://www.rfc-editor.org/info/rfc8529>.

[RFC9020] Litkowski, S., Qu, Y., Lindem, A., Sarkar, P., and J. Tantsura, "YANG Data Model for Segment Routing", RFC 9020, DOI 10.17487/RFC9020, May 2021, <https://www.rfc-editor.org/info/rfc9020>.

[RFC9067] Qu, Y., Tantsura, J., Lindem, A., and X. Liu, "A YANG Data Model for Routing Policy", RFC 9067, DOI 10.17487/RFC9067, October 2021, <https://www.rfc-editor.org/info/rfc9067>.

[I-D.ietf-idr-bgp-model] Jethanandani, M., Patel, K., Hares, S., and J. Haas, "YANG Model for Border Gateway Protocol (BGP-4)", Work in Progress, Internet-Draft, draft-ietf-idr-bgp-model-17, 5 July 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-idr-bgp-model-17>.

[I-D.ietf-spring-srv6-yang] Raza, S., Agarwal, S., Liu, X., Hu, Z., Hussain, I., Shah, H. C., Voyer, D., Matsushima, S., Horiba, K., Rajamanickam, J., and A. Abdelsalam, "YANG Data Model for SRv6 Base and Static", Work in Progress, Internet-Draft, draft-ietf-spring-srv6-yang-02, 23 September 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-spring-srv6-yang-02>.

[I-D.mpmz-bess-mup-safi] Murakami, T., Patel, K., Matsushima, S., Zhang, Z. J., Agrawal, S., and D. Voyer, "BGP Extensions for the Mobile User Plane (MUP) SAFI", Work in Progress, Internet-Draft, draft-mpmz-bess-mup-safi-02, 13 March 2023, <https://datatracker.ietf.org/doc/html/draft-mpmz-bess-mup-safi-02>.

[I-D.mhkk-dmm-srv6mup-architecture] Matsushima, S., Horiba, K., Khan, A., Kawakami, Y., Murakami, T., Patel, K., Kohno, M., Kamata, T., Camarillo, P., Horn, J., Voyer, D., Zadok, S., Meilik, I., Agrawal, A., and K. Perumal, "Mobile User Plane Architecture using Segment Routing for Distributed Mobility Management", Work in Progress, Internet-Draft, draft-mhkk-dmm-srv6mup-architecture-05, 13 March 2023, <https://datatracker.ietf.org/doc/html/draft-mhkk-dmm-srv6mup-architecture-05>.

7.2. Informative References

Appendix A. Complete Tree Diagram

Here is a complete tree diagram for the configuration and operational part of the model.

```
module: ietf-mup

  augment /rt:routing/rt:control-plane-protocols
            /rt:control-plane-protocol/bgp:bgp/bgp:global
            /bgp:afi-safis/bgp:afi-safi:
    +--rw ipv4-mup
    |  +--rw rts
    |  |  +--rw rt* [name]
    |  |     +--rw name              identityref
    |  |     +--rw route-policy?     leafref
    |  |     +--rw route-targets
    |  |        +--rw route-target* [target type]
    |  |           +--rw target    rt-types:route-target
    |  |           +--rw type      rt-types:route-target-type
    |  +--rw routing-table-limit
    |  |  +--rw routes* [type]
    |  |     +--rw type              identityref
    |  |     +--rw number?           uint32
    |  |     +--rw (action)?
    |  |        +--:(percent)
    |  |        |  +--rw percent?   rt-types:percentage
    |  |        +--:(simple)
    |  |           +--rw simple?    boolean
    |  +--rw segment* [type]
    |  |  +--rw type            identityref
    |  |  +--rw locator?        leafref
    |  |  +--rw entry*          union
    |  |  +--rw mup-ext-comm*   mup-ext-community-type
    |  +--rw architecture-type?     identityref
    +--rw ipv6-mup
       +--rw rts
       |  +--rw rt* [name]
       |     +--rw name              identityref
       |     +--rw route-policy?     leafref
       |     +--rw route-targets
       |        +--rw route-target* [target type]
       |           +--rw target    rt-types:route-target
       |           +--rw type      rt-types:route-target-type
       +--rw routing-table-limit
       |  +--rw routes* [type]
       |     +--rw type              identityref
```

```
          |     +--rw number?          uint32
          |     +--rw (action)?
          |        +--:(percent)
          |        |  +--rw percent?   rt-types:percentage
          |        +--:(simple)
          |           +--rw simple?    boolean
          +--rw segment* [type]
          |  +--rw type             identityref
          |  +--rw locator?         leafref
          |  +--rw entry*           union
          |  +--rw mup-ext-comm*    mup-ext-community-type
          +--rw architecture-type?     identityref
    augment /rt:routing/rt:control-plane-protocols
              /rt:control-plane-protocol/bgp:bgp/bgp:global
              /bgp:afi-safis/bgp:afi-safi/bgp:ipv4-unicast:
      +--rw rts
         +--rw rt* [name]
            +--rw name              identityref
            +--rw route-policy?     leafref
            +--rw route-targets
               +--rw route-target* [target type]
                  +--rw target    rt-types:route-target
                  +--rw type      rt-types:route-target-type
    augment /rt:routing/rt:control-plane-protocols
              /rt:control-plane-protocol/bgp:bgp/bgp:global
              /bgp:afi-safis/bgp:afi-safi/bgp:ipv6-unicast:
      +--rw rts
         +--rw rt* [name]
            +--rw name              identityref
            +--rw route-policy?     leafref
            +--rw route-targets
               +--rw route-target* [target type]
                  +--rw target    rt-types:route-target
                  +--rw type      rt-types:route-target-type
    augment /rt:routing/rt:control-plane-protocols
              /rt:control-plane-protocol/bgp:bgp/bgp:global
              /bgp:afi-safis/bgp:afi-safi/bgp:l3vpn-ipv4-unicast:
      +--rw rts
         +--rw rt* [name]
            +--rw name              identityref
            +--rw route-policy?     leafref
            +--rw route-targets
               +--rw route-target* [target type]
                  +--rw target    rt-types:route-target
                  +--rw type      rt-types:route-target-type
    augment /rt:routing/rt:control-plane-protocols
              /rt:control-plane-protocol/bgp:bgp/bgp:global
              /bgp:afi-safis/bgp:afi-safi/bgp:l3vpn-ipv6-unicast:
```

```
      +--rw rts
         +--rw rt* [name]
            +--rw name             identityref
            +--rw route-policy?    leafref
            +--rw route-targets
               +--rw route-target* [target type]
                  +--rw target    rt-types:route-target
                  +--rw type      rt-types:route-target-type
    augment /rt:routing/rt:control-plane-protocols
              /rt:control-plane-protocol/bgp:bgp/bgp:neighbors
              /bgp:neighbor/bgp:afi-safis/bgp:afi-safi:
      +--rw ipv4-mup!
      +--rw ipv6-mup!
    augment /rt:routing/rt:control-plane-protocols
              /rt:control-plane-protocol/bgp:bgp/bgp:neighbors
              /bgp:neighbor/bgp:statistics:
      +--ro isd-sent?             yang:zero-based-counter32
      +--ro isd-received?         yang:zero-based-counter32
      +--ro dsd-sent?             yang:zero-based-counter32
      +--ro dsd-received?         yang:zero-based-counter32
      +--ro type-1-st-sent?       yang:zero-based-counter32
      +--ro type-1-st-received?   yang:zero-based-counter32
      +--ro type-2-st-sent?       yang:zero-based-counter32
      +--ro type-2-st-received?   yang:zero-based-counter32
    augment /rt:routing/rt:control-plane-protocols
              /rt:control-plane-protocol/bgp:bgp/bgp:global:
      +--rw srv6
         +--rw locator?                  leafref
         +--ro sid-manager-connected?    boolean
         +--ro locator-registered?       boolean
         +--ro micro-segment-enabled?    boolean
         +--rw sid-allocation-mode?      identityref
         +--rw mobile
            +--rw encapsulation
            |  +--rw locator?   -> ../../../locator
            +--rw decapsulations
               +--rw decapsulation* [id]
                  +--rw id                      uint16
                  +--rw locator?                -> ../../../../locator
                  +--rw mup-ext-communities
                     +--rw mup-ext-community*   mup-ext-community-type
    augment /ni:network-instances/ni:network-instance/ni:ni-type:
      +--:(mup)
         +--rw mup
            +--rw rd
               +--rw rd?        union
               +--ro auto-rd?   rt-types:route-distinguisher
    augment /rt-pol:routing-policy/rt-pol:policy-definitions
```

```
               /rt-pol:policy-definition/rt-pol:statements
               /rt-pol:statement/rt-pol:conditions/bp:bgp-conditions:
       +--rw match-mup
          +--rw route-type?   identityref
     augment /rt-pol:routing-policy/rt-pol:policy-definitions
               /rt-pol:policy-definition/rt-pol:statements
               /rt-pol:statement/rt-pol:actions:
       +--rw mup-actions
          +--rw accept-route?   boolean
```

Figure 3: Complete tree diagram

Appendix B. Configuration examples

This section documents some example configurations for MUP.

B.1. MUP configuration under BGP

```
<!--
    This example shows a MUP configuration, with routing policy
    configured for route target.

draft-ietf-spring-srv6-yang defines srv6 locators
under /routing/segment-routing/srv6 path, instead of
putting them under BGP.
-->

<?xml version="1.0" encoding="UTF-8"?>
<routing-policy xmlns="urn:ietf:params:xml:ns:yang:ietf-routing-policy">
  <policy-definitions>
    <policy-definition>
      <name>route-target-policy</name>
      <statements>
        <statement>
          <name>10</name>
          <conditions>
            <bgp-conditions xmlns="urn:ietf:params:xml:ns:yang:ietf-bgp-policy">
              <match-mup xmlns="urn:ietf:params:xml:ns:yang:ietf-mup">
                <route-type>type-1-st</route-type>
              </match-mup>
            </bgp-conditions>
          </conditions>
          <actions xmlns="urn:ietf:params:xml:ns:yang:ietf-routing-policy">
            <bgp-actions xmlns="urn:ietf:params:xml:ns:yang:ietf-bgp-policy">
              <set-ext-community>
                <communities>route-target:100:5000</communities>
                <options>add</options>
              </set-ext-community>
```

```
          </bgp-actions>
          <mup-actions xmlns="urn:ietf:params:xml:ns:yang:ietf-mup">
            <accept-route>true</accept-route>
          </mup-actions>
        </actions>
      </statement>
    </statements>
  </policy-definition>
</policy-definitions>
</routing-policy>

<routing
    xmlns="urn:ietf:params:xml:ns:yang:ietf-routing"
    xmlns:bt="urn:ietf:params:xml:ns:yang:iana-bgp-types"
    xmlns:mup="urn:ietf:params:xml:ns:yang:ietf-mup"
    xmlns:srv6="urn:ietf:params:xml:ns:yang:ietf-srv6-base">
  <segment-routing
      xmlns="urn:ietf:params:xml:ns:yang:ietf-segment-routing">
    <srv6
        xmlns="urn:ietf:params:xml:ns:yang:ietf-srv6-base">
      <locators>
        <locator>
          <name>mup-gw-1</name>
          <prefix>
            <address>bead:1000::</address>
            <length>32</length>
          </prefix>
        </locator>
        <locator>
          <name>mup-gw-2</name>
          <prefix>
            <address>bead:1010::</address>
            <length>32</length>
          </prefix>
        </locator>
      </locators>
    </srv6>
  </segment-routing>
  <control-plane-protocols>
    <control-plane-protocol>
      <type
          xmlns:bgp="urn:ietf:params:xml:ns:yang:ietf-bgp">bgp:bgp</type>
      <name>b1</name>
      <bgp
          xmlns="urn:ietf:params:xml:ns:yang:ietf-bgp">
        <global>
          <as>64496</as>
          <identifier>11.11.11.11</identifier>
```

```
          <afi-safis>
            <afi-safi>
              <name
                  xmlns:mup="urn:ietf:params:xml:ns:yang:ietf-mup">mup:ipv4-mup</
name>
              <ipv4-mup xmlns="urn:ietf:params:xml:ns:yang:ietf-mup">
                <rts>
                  <rt>
                    <name>ipv4-mup</name>
                    <route-policy>route-target-policy</route-policy>
                    <route-targets>
                      <route-target>
                        <target>100:4000</target>
                        <type>import</type>
                      </route-target>
                    </route-targets>
                  </rt>
                </rts>
              </ipv4-mup>
            </afi-safi>
          </afi-safis>
        </global>
        <neighbors>
          <neighbor>
            <remote-address>33.33.33.33</remote-address>
            <afi-safis>
              <afi-safi>
                <name
                    xmlns:mup="urn:ietf:params:xml:ns:yang:ietf-mup">mup:ipv4-mup
</name>
              </afi-safi>
            </afi-safis>
          </neighbor>
        </neighbors>
      </bgp>
    </control-plane-protocol>
  </control-plane-protocols>
</routing>
```

Figure 4: MUP configuration under BGP

B.2. Example MUP configuration of SRv6.

```
<!--
    This example shows a MUP configuration.
-->

<?xml version="1.0" encoding="UTF-8"?>
<interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"
            xmlns:ipv4="urn:ietf:params:xml:ns:yang:ietf-ip"
```

```
            xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type">
  <interface>
    <name>loopback0</name>
    <type>ianaift:softwareLoopback</type>
    <ipv4>
        xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
      <address>
        <ip>11.11.11.11</ip>
        <prefix-length>32</prefix-length>
      </address>
    </ipv4>
  </interface>
  <interface>
    <name>swp10</name>
    <type>ianaift:ethernetCsmacd</type>
    <ipv4
        xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
      <address>
        <ip>20.1.1.1</ip>
        <prefix-length>24</prefix-length>
      </address>
    </ipv4>
  </interface>
</interfaces>


<routing
    xmlns="urn:ietf:params:xml:ns:yang:ietf-routing"
    xmlns:bt="urn:ietf:params:xml:ns:yang:ietf-bgp-types"
    xmlns:mup="urn:ietf:params:xml:ns:yang:ietf-mup"
    xmlns:srv6="urn:ietf:params:xml:ns:yang:ietf-srv6-base">
  <segment-routing
      xmlns="urn:ietf:params:xml:ns:yang:ietf-segment-routing">
    <srv6
        xmlns="urn:ietf:params:xml:ns:yang:ietf-srv6-base">
      <locators>
        <locator>
          <name>mup-gw-1</name>
          <prefix>
            <!-- draft-ietf-spring-srv6-yang defines address
                 as ip-address instead of ip-prefix, which
                 would allow this to be specified the following
                 address as bead:1000::/48.
            -->
            <address>bead:1000::</address>
            <!-- draft-ietf-spring-srv6-yang defines length
                 to be prefix length, instead of sid allocation
                 length. Additionally, it puts a constraint of
```

```
                 range as 32..98, which means this value cannot
                 be 16.
            -->
            <length>32</length>
          </prefix>
        </locator>
        <locator>
          <name>mup-gw-2</name>
          <prefix>
            <address>bead:1010::</address>
            <length>32</length>
          </prefix>
        </locator>
      </locators>
    </srv6>
  </segment-routing>
  <control-plane-protocols>
    <control-plane-protocol>
      <type
          xmlns:bgp="urn:ietf:params:xml:ns:yang:ietf-bgp">bgp:bgp</type>
      <name>b1</name>
      <bgp
          xmlns="urn:ietf:params:xml:ns:yang:ietf-bgp">
        <global>
          <as>64496</as>
          <identifier>11.11.11.11</identifier>
          <!-- draft-ietf-idr-bgp-yang does not define
               route-distinguisher at a global level.

               draft-ietf-sprint-ipv6-yang does not define a
               way to reference a srv6 locator from within a
               BGP instance.

              Additionally, it does not allow a SID allocation
              mode for something like 'instance-sid' or
              'prefix-sid'
          -->
          <afi-safis>
            <afi-safi>
              <name
                  xmlns:mup="urn:ietf:params:xml:ns:yang:ietf-mup">mup:ipv4-mup</
name>
            </afi-safi>
          </afi-safis>
        </global>
        <neighbors>
          <neighbor>
            <remote-address>33.33.33.33</remote-address>
            <afi-safis>
```

```
              <afi-safi>
                <name
                    xmlns:mup="urn:ietf:params:xml:ns:yang:ietf-mup">mup:ipv4-mup
</name>
              </afi-safi>
            </afi-safis>
          </neighbor>
        </neighbors>
      </bgp>
    </control-plane-protocol>
  </control-plane-protocols>
</routing>
```

Figure 5: Example MUP configuration in BGP for SRv6

B.3. Example MUP configuration for RT.

```
<!--
    This example shows a MUP configuration.
-->

<?xml version="1.0" encoding="UTF-8"?>
<interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"
            xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type">
  <interface>
    <name>loopback0</name>
    <type>ianaift:softwareLoopback</type>
    <ipv4
        xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
      <address>
        <ip>11.11.11.11</ip>
        <prefix-length>32</prefix-length>
      </address>
    </ipv4>
  </interface>
  <interface>
    <name>swp10</name>
    <type>ianaift:ethernetCsmacd</type>
    <ipv4
        xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
      <address>
        <ip>20.1.1.1</ip>
        <prefix-length>24</prefix-length>
      </address>
    </ipv4>
  </interface>
</interfaces>

<routing
```

```
    xmlns="urn:ietf:params:xml:ns:yang:ietf-routing"
    xmlns:bt="urn:ietf:params:xml:ns:yang:ietf-bgp-types"
    xmlns:mup="urn:ietf:params:xml:ns:yang:ietf-mup"
    xmlns:srv6="urn:ietf:params:xml:ns:yang:ietf-srv6-base">
  <segment-routing
      xmlns="urn:ietf:params:xml:ns:yang:ietf-segment-routing">
    <srv6
        xmlns="urn:ietf:params:xml:ns:yang:ietf-srv6-base">
      <locators>
        <locator>
          <name>mup-gw-1</name>
          <prefix>
            <!-- draft-ietf-spring-srv6-yang defines address
                 as ip-address instead of ip-prefix, which
                 would allow this to be specified the following
                 address as bead:1000::/48.
            -->
            <address>bead:1000::</address>
            <length>32</length>
          </prefix>
        </locator>
        <locator>
          <name>mup-gw-2</name>
          <prefix>
            <!-- draft-ietf-spring-srv6-yang defines address
                 as ip-address instead of ip-prefix, which
                 would allow this to be specified the following
                 address as bead:1000::/48.
            -->
            <address>bead:1010::</address>
            <length>32</length>
          </prefix>
        </locator>
      </locators>
    </srv6>
  </segment-routing>
  <control-plane-protocols>
    <control-plane-protocol>
      <type
          xmlns:bgp="urn:ietf:params:xml:ns:yang:ietf-bgp">bgp:bgp</type>
      <name>b1</name>
      <bgp
          xmlns="urn:ietf:params:xml:ns:yang:ietf-bgp">
        <global>
          <as>64496</as>
          <identifier>11.11.11.11</identifier>
          <!--
              draft-ietf-spring-ipv6-yang does not define a
```

```
              way to reference a srv6 locator from within a
              BGP instance.

              Cannot configure RD in IETF models as RD is configured
              at a NI level, and configuring anything at the NI level
              requires support of schema mount which most tools do not
              support.
          -->
          <srv6
              xmlns="urn:ietf:params:xml:ns:yang:ietf-mup">
            <locator>mup-gw-1</locator>
            <sid-allocation-mode>instance-sid</sid-allocation-mode>
          </srv6>
          <afi-safis>
            <afi-safi>
              <name
                  xmlns:mup="urn:ietf:params:xml:ns:yang:ietf-mup">mup:ipv4-mup</
name>
              <ipv4-mup
                  xmlns="urn:ietf:params:xml:ns:yang:ietf-mup">
                <rts>
                  <rt>
                    <name>ipv4-mup</name>
                    <route-targets>
                      <route-target>
                        <target>100:6000</target>
                        <type>import</type>
                      </route-target>
                      <route-target>
                        <target>100:4000</target>
                        <type>export</type>
                      </route-target>
                    </route-targets>
                  </rt>
                </rts>
                &lt;segment&gt;
                  <type>isd</type>
                  <locator>mup-gw-2</locator>
                  <entry>swp10</entry>
                &lt;/segment&gt;
              </ipv4-mup>
            </afi-safi>
          </afi-safis>
        </global>
      </bgp>
    </control-plane-protocol>
  </control-plane-protocols>
</routing>
```

Figure 6: Example MUP configuration in BGP for RT

Acknowledgements

TBA

Contributors

Thanks to all of the contributors.

Authors' Addresses

Mahesh Jethanandani (editor)
Arrcus, Inc
Email: mjethanandani@gmail.com

Tetsuya Murakami
Arrcus, Inc
Email: tetsuya@arrcus.com

Kalyani Rajaram
Arrcus, Inc
Email: kalyanir@arrcus.com

Satoru Matsushima
SoftBank
Email: satoru.matsushima@g.softbank.co.jp