Spring WG
Internet-Draft
Intended status: Standards Track
Expires: September 6, 2015

Fangwei. Hu
Ran. Chen
Frank. Feng
ZTE Corporation
March 5, 2015

YANG Data model for Segment Routing
draft-hu-spring-yang-00.txt

Abstract

This document defines a YANG data model for segment routing technology. The data model covers configuration data and event notifications.

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 http://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 September 6, 2015.

Copyright Notice

Copyright (c) 2015 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 (http://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. Design of Data model 2
3. Spring YANG Data model 4
4. Spring YANG notification 11
5. Security Considerations 13
6. Acknowledgements 13
7. IANA Considerations 13
8. Normative References 13
Authors' Addresses 13

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 YANG [RFC6020] data model for the management of SPRING(Source Packet Routing in Networking) [I-D.ietf-spring-segment-routing]. The YANG data design, YANG data model and YANG notification event are introduced in this proposal.

2. Design of Data model

This document proposes a base segment routing YANG data model. The module can be augmented for other segment routing extended features with their specific definitions, such as segment routing OAM, IPv6 segment routing, etc. In addition, a notification event for the segment routing is defined.

The figure below describe the overall structure of the Segment Routing Yang model:

```
module: Spring
 +--rw global-block
 |   +--rw Min
 |   +--rw Max
 +--rw segment
 |   +--rw name
 |   +--rw sid
 |   +--rw type
 |   +--rw scope
 |   +--rw prefix-flag
 |      +--rw R
 |      +--rw N
 |      +--rw P
 |      +--rw E
```

```
    |      +--rw V
    |      +--rw L
    |   +--rw adjacency-flag
    |      +--rw F
    |      +--rw B
    |      +--rw V
    |      +--rw L
    |      +--rw S
    +--rw fec-mapping
    |   +--fec-prefix
    |   +--explicit-route
    |   +--sid-label-binding-flag
    |      +--rw F
    |      +--rw M
    |      +--rw S
    |      +--rw D
    |      +--rw A
    +--rw label-entity
    |   +--rw sid
    |   +--rw incoming
    |   +--rw outgoing
    |   +--rw next-hop
    |   +--rw header-operation
    |   +--rw egress-interface
    +--rw tunnel
    |   +--rw name
    |   +--rw ingress
    |   +--rw egress
    |   +--rw priority
    |   +--rw explicit-route
    |   +--rw path-type
    |   +--rw frr-protection-method
    +


   notifications:
    +--n tunnel-event
    |  +--ro name
    |  +--ro tunnel-creation
    |  +--ro tunnel-deletion
    |  +--ro tunnel-state
    |  +--ro frr-status-change
    |  +--ro path-protection-status-change
    +
```

3. Spring YANG Data model

```
 module ietf-spring {

    namespace "urn:ietf:params:xml:ns:yang:ietf-spring";

    prefix "Spring";

    organization
        "IETF SPRING (Source Packet Routing in Networking) Working Group";

      contact
        "WG Web:   <http://tools.ietf.org/wg/spring/>
         WG List:  <mailto:spring@ietf.org>

         WG Chair: Bruno Decraene
                   <mailto:bruno.decraene@orange.com>

         WG Chair: John Scudder
                   <mailto:jgs@juniper.net>
         Editor:   Fangwei Hu
                   <mailto:hu.fangwei@zte.com.cn>

         Editor:   Ran Chen
                   <mailto:chen.ran@zte.com.cn>

         Editor:   Frank Feng
                   <mailto:feng.chong33@zte.com.cn>";

    description
      "The YANG module defines data model for the management of Spring";

    revision 2015-03-05 {
      description
        "Initial revision";
    }

    //typedefs
    typedef sid {
        description
          "The type of segment index.";
        type uint32;
    }

    typedef tunnel-state {
     type enumeration {
      enum up;
      enum down;
```

```
     }
    }

    typedef frr-status {
     type enumeration {
      enum master;
      enum slave;
     }
    }

    typedef path-protection-status {
     type enumeration {
      enum node;
      enum link;
      }
    }

    //data defs
    container global-block {
     description
       "SID global block, the set of local
        labels reserved for global segments";

    leaf min {
     type uint32;
     description
        "The begin of the SID global block";
     }

    leaf max {
     type uint32;
     description
        "The end of the SID global block";
     }
    }

    list segment {
     key name;
     unique sid;
     leaf name {
     description
       "identify of a segment.";
     type string;
     }

     leaf sid {
     description
       "index of a segment.";
```

```
     config false;
     type sid;
    }

    leaf type {
     description
        "The type of a segment. A segment could be a prefix, node,
         adjacency, and anycast segment or other types defined in the future.";
     mandatory true;

    type enumeration {
      enum prefix;
      enum node;
      enum adjacency;
      enum anycast;
     }
    }

    leaf scope {
      description
        "The scope of segment";
      mandatory true;
      type enumeration {
      enum global;
      enum local;
     }
    }

    leaf prefix-flag {
      description
        "The flag of Prefix Segment ";
      when "type=prefix";
      type bits {
      bit R {
        description
          "Re-advertisement flag, If set, then the prefix to which this
           Prefix-SID is attached, has been propagated by the router
           either from another level (i.e.: from level-1 to level-2 or
           the opposite) or from redistribution (e.g.: from another
           protocol)";
        position 1;
        }

      bit N {
        description
          "Node-SID flag, If set, then the Prefix-SID refers to the router
           identified by the prefix";
        position 2;
```

```
        }

      bit P {
        description
          "no-PHP flag, If set, then the penultimate hop MUST NOT pop the
           Prefix-SID before delivering the packet to the node that advertised
           the Prefix-SID";
        position 3;
        }

      bit E {
        description
          "Explicit-Null Flag, If set, any upstream neighbor of the Prefix-SID
          originator MUST replace the Prefix-SID with a Prefix-SID having an
          Explicit-NULL value (0 for IPv4 and 2 for IPv6) before forwarding
          the packet";
        position 4;
        }

      bit V {
        description
          "Value flag, If set, then the Prefix-SID carries a value (instead of
           an index).";
        position 5;
        }

      bit L {
        description
          "Local Flag, If set, then the value/index carried by the Prefix-SID
           has local significance";
        position 6;
        }
      }
    }

    leaf adjacency-flag {
     description
        "The flag of Adjacency Segment";
     when "type=adjacency";

     type bits {
      bit F {
        description
          "Address-Family flag. If unset, then the Adj-SID refers to an
           adjacency with outgoing IPv4 encapsulation.  If set then the
           Adj-SID refers to an adjacency with outgoing IPv6 encapsulation";
        position 1;
        }
```

```
      bit B {
        description
          "Backup flag. If set, the Adj-SID refers to an adjacency being
           protected.";
        position 2;
        }

      bit V {
        description
          "If set, then the Adj-SID carries a value.";
        position 3;
       }

      bit L {
        description
          "If set, then the value/index carried by the Adj-SID has local
           significance.";
        position 4;
       }

      bit S {
        description
          "Set Flag. When set, the S-Flag indicates that the Adj-SID refers
           to a set of adjacencies.";
        position 5;
       }
     }
    }


   }

   list fec-mapping {
     key fec-prefix;
     leaf fec-prefix {
     description
       "The prefix address of FEC of the router.";
     type uint32;
     }

     leaf-list explicit-route {
       type sid;
     }

     leaf sid-label-binding-flag {
       description
         "The flag of SID/Label Binding ";
       type bits {
```

```
         bit F {
           description
             "Address Family flag. If unset, then the Prefix FEC carries
              an IPv4 Prefix. If set, then the Prefix FEC carries an IPv6
Prefix";
           position 1;
          }

         bit M {
           description
             "Mirror Context flag. Set if the advertised SID/path corresponds
              to a mirrored context.";
           position 2;
          }

         bit S {
          description
             "If set, the SID/Label Binding TLV SHOULD be flooded across the
              entire routing domain. If the S flag is not set, the SID/Label
              Binding TLV MUST NOT be leaked between levels ";
          position 3;
         }

         bit D {
          description
             "when the SID/Label Binding TLV is leaked from level-2 to level-1,
              the D bit MUST be set. Otherwise, this bit MUST be clear.SID/
Label
              Binding TLVs with the D bit set MUST NOT be leaked from level-1
to
              level-2 ";
          position 4;
         }

         bit A {
          description
             "The originator of the SID/Label Binding TLV MAY set the A bit in
              order to signal that the prefixes and SIDs advertised in the
              SID/Label Binding TLV are directly connected to their
originators.";
              position 5;
         }
       }
     }
  }

   list label-entity {
     key sid;
```

```
     config false;
     leaf sid {
      description
        "The index of a segment.";
```

```
      type sid;
     }

     leaf incoming {
       description
         "Incoming active segment.";
       type sid;
      }

     leaf outgoing {
       description
         "Outgoing active segment.";
       type sid;
      }

     leaf next-hop {
       description
         "The IP address of the next hop for the continue operation.";
       type uint32;
      }

     leaf header-operation {
       description
         "The operation action for the flow.";
       type enumeration {
         enum push;
         enum next;
         enum continue;
        }
     }

     leaf egress-interface {
       description
         "The outgoing interface for the flow.";
       type uint32;
      }
   }


   list tunnel {
     description
       "The spring tunnel";
     key name;
     leaf name {
     type string;
    }

     leaf ingress {
```

```
      description
        "Ingress SID";
      type sid;
    }

    leaf egress {
      description
        "Egress SID";
      type sid;
    }

    leaf priority {
      description
        "The priority of tunnel.";
      type uint32;
    }

    leaf-list explicit-route {
       type sid;
    }

    leaf path-type {
      description
        "Whether the tunnel is a master or slave path?";
      type frr-status;
    }

    leaf frr-protection-method {
      description
        "whether the protection object is a node or link?";
      type path-protection-status;
    }
  }
```

4. Spring YANG notification

```
      //notifications
      notification tunnels-event {
        list tunnel {
          key name;
          leaf name {
            description
              "The name of tunnel";
            type string;
           }

          choice event {
           mandatory true;
```

```
            case creation {
              leaf creation {
              type empty;
             }
            }
            case deletion {
             leaf deletion {
             type empty;
             }
            }

            case state-transition {
             container state-transition {
              must "from != to";
              leaf from {
                type tunnel-state;
                mandatory true;
               }

              leaf to {
                type tunnel-state;
                mandatory true;
               }
             }
            }

            case frr-status-change {
             container frr-status-change {
              must "from != to";
              leaf from {
               type frr-status;
               mandatory true;
              }

              leaf to {
               type frr-status;
               mandatory true;
              }
             }
            }

            case path-protection-status-change {
             container path-protection-status-change {
              must "from != to";
               leaf from {
                 type path-protection-status;
                 mandatory true;
               }
```

```
              leaf to {
                type path-protection-status;
                mandatory true;
              }
             }
            }
          }
        }
      }
```

5. Security Considerations

TBD.

6. Acknowledgements

TBD.

7. IANA Considerations

TBD.

8. Normative References

[I-D.ietf-spring-segment-routing]
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., Shakir, R., Tantsura, J., and E. Crabbe, "Segment Routing Architecture", draft-ietf-spring-segment-routing-01 (work in progress), February 2015.

[RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

[RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and A. Bierman, "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.

Authors' Addresses

Fangwei Hu
ZTE Corporation
No.889 Bibo Rd
Shanghai 201203
China

Phone: +86 21 68896273
Email: hu.fangwei@zte.com.cn

Ran Chen
ZTE Corporation
No.50 Software Avenue,Yuhuatai District
Nanjing, Jiangsu Province 210012
China

Phone: +86 025 88014636
Email: chen.ran@zte.com.cn

Frank Feng
ZTE Corporation
No.86 Zijinghua Rd
Nanjing, Jiangsu Province 210012
China

Phone: +86 21 68896273
Email: feng.chong33@zte.com.cn