A YANG Data Model for Segment Routing
draft-ietf-spring-sr-yang-07

Stephane Litkowski (stephane.litkowski@orange.com)
Yingzhen Qu (yingzhen.qu@huawei.com)
Pushpasis Sarkar (pushpasis.ietf@gmail.com)
Jeff Tantsura (jefftant.ietf@gmail.com)
module: ietf-segment-routing
augment /rt:routing:
  +rw segment-routing
    +rw transport-type? identityref
    +ro node-capabilities
      | +ro transport-planes* [transport-plane]
      | | +ro transport-plane identityref
      | | +ro readable-label-stack-depth? uint8
    +rw msd {msd}?
    | ....
  +rw bindings
    | +rw mapping-server {mapping-server}?
    | | +rw policy* [name]
    | | | +rw name string
    | | | +rw ipv4
    | | | | +rw mapping-entry* [prefix algorithm]
    | | | | | ....
    | | | +rw ipv6
    | | | | | | +rw mapping-entry* [prefix algorithm]
    | | | | | | | ....
    | | | +rw ipv4
    | | | | | | | | +rw prefix-sid-map
    | | | | | | | | | ....
    | | | +rw ipv6
    | | | | | | | | | +rw prefix-sid-map
    | | | | | | | | | | ....
    | | | | +ro sid-list
    | | | | | +ro sid* [target sid source
      | | | | | | source-protocol binding-type]
    | | | | | | | +ro target string
    | | | | | | | | +ro sid uint32
    | | | | | | | | +ro algorithm? uint8
    | | | | | | | | | +ro source inet:ip-address
    | | | | | | | | | | +ro used? boolean
    | | | | | | | | | | | +ro source-protocol -> /rt:routing
    | | | | | | | | | | | | +control-plane-protocols
    | | | | | | | | | | | | | +control-plane-protocol
    | | | | | | | | | | | | | | +name
    | | | | | | | | | | | | | | | +ro binding-type enumeration
    | | | | | | | | | | | | | | | | +ro scope? enumeration
    | | | | | +ro label-blocks*
### Segment Routing Global Block

- Defines a list of label blocks represented by a pair of lower-bound/upper-bound labels.

```yaml
module: ietf-segment-routing
augment /rt:routing:
  | +--rw global-srgb
  |     | +--rw srgb* [lower-bound upper-bound]
  |     |     | +--rw lower-bound       uint32
  |     |     |     | +--rw upper-bound       uint32
```

```yaml
grouping srgb-cfg {
  description
  "Grouping for SR Label Range configuration."
  list srgb {
    key "lower-bound upper-bound"
    ordered-by user;
    description
    "List of global blocks to be advertised."
    uses srlr;
  }
}
```

```yaml
feature protocol-srgb {
  description
  "Support per-protocol srgb configuration."
}
```

```yaml
container global-srgb {
  description
  "Global SRGB configuration."
  uses sr-cmn:srgb-cfg;
}
```
Segment Routing Local Block (SRLB)

- Defines a list of label blocks represented by a pair of lower-bound/upper-bound labels, reserved for local SIDs.

```
grouping srlb-cfg {
  description
    "Grouping for SR Local Block range configuration.";
  list srlb {
    key "lower-bound upper-bound";
    ordered-by user;
    description
      "List of SRLBs.";
    uses srlr;
  }
}
container srlb {
  description
    "SR Local Block configuration.";
  uses sr-cmn:srlb-cfg;
}
```

```
augment /rt: routing:
  +--rw segment-routing
     | ....
     +--rw srlb
        | +--rw srlb [{lower-bound upper-bound}]
        |     +--rw lower-bound  uint32
        |     +--rw upper-bound  uint32
```
Maximum SID Depth (MSD)

```text
feature msd {
  description
  "Support of signaling MSD (Maximum SID Depth) in IGP.";
}
grouping msd-cfg {
  description
  "MSD configuration grouping.";
  leaf node-msd {
    type uint8;
    description
    "Node MSD is the lowest MSD supported by the node.";
  }
}
container link-msd {
  description
  "Link MSD is a number representing the particular link MSD value.";
  list link-msds {
    key "interface";
    description
    "List of link MSDs.";
    leaf interface {
      type if:interface-ref;
      description
      "Name of the interface.";
    }
  }
}
container msd {
  if-feature "msd";
  description
  "MSD configuration.";
  uses msd-cfg;
}
```

module: ietf-segment-routing
augment /rt:routing:
  +--rw msd {msd}?
    |   +--rw node-msd?  uint8
    |   +--rw link-msd
    |   |   +--rw link-msds* [interface]
    |   |   |   +--rw interface if:interface-ref
    |   |   |   +--rw msd?  uint8
```
Notifications

+---n segment-routing-global-srgb-collision
  | +--ro srgb-collisions*  uint32
  | +--ro lower-bound?     uint32
  | +--ro upper-bound?     uint32
  | +--ro routing-protocol? -> /rt:routing/control-plane-protocols
  |                        /control-plane-protocol/name
  | +--ro originating-rtr-id? router-id
+---n segment-routing-global-sid-collision
  | +--ro received-target?  string
  | +--ro new-sid-rtr-id?   router-id
  | +--ro original-target?  string
  | +--ro original-sid-rtr-id? router-id
  | +--ro index?           uint32
  | +--ro routing-protocol? -> /rt:routing/control-plane-protocols
  |                        /control-plane-protocol/name
+---n segment-routing-index-out-of-range
  +--ro received-target?  string
  +--ro received-index?   uint32
  +--ro routing-protocol? -> /rt:routing/control-plane-protocols
                           /control-plane-protocol/name
Transport protocols have been defined using identities. When SRv6 is ready, a separate module will be defined to augment this base model.
Next Steps

• Will do an update after this IETF
• Collect/address comments
• WGLC soon
Question?

Thank You