

draft-litkowski-isis-yang-isis-cfg

IETF 90 - Toronto

S. Litkowski, Orange

Introduction

- Goal :
 - Define a standardized model for ISIS configuration and operation
- Use case :
 - Unified provisioning system between vendors
 - Unified way to collect ISIS protocol informations (computation times, events ...)
- As a SP, we are clearly looking at such solution !

Introduction

- Was first published as draft-litkowski-netmod-isis-cfg
- Netmod chairs and AD decided that it would be better to manage the draft in ISIS WG (where ISIS experts are) and YANG doctor would help for YANG definition
 - Draft renamed to draft-litkowski-isis-yang-isis-cfg
- Draft is inline with ietf-routing definitions (CORE routing model)

Main trees : configuration

```
augment
  /rt:routing/rt:routing-instance/rt:routing-protocols/rt:routing-protocol:
    +-rw isis
      +-rw isis-level?          isis-level
      +-rw nsap-address        simple-iso-address
      +-rw ipv4-router-id?    inet:ipv4-address
      +-rw ipv6-router-id?    inet:ipv6-address
      +-rw reference-bandwidth? uint32
      +-rw lsp-mtu?           uint16
      +-rw lsp-lifetime?      uint16
      +-rw lsp-refresh?       uint16
      +-rw psnp-authentication? boolean
      +-rw csnp-authentication? boolean
      +-rw hello-authentication? boolean
      +-rw authentication-key? string
      +-rw authentication-type? enumeration
      +-rw isis-multi-topology-cfg
        | +-rw ipv4-unicast?   boolean
        | +-rw ipv6-unicast?   boolean
        | +-rw ipv4-multicast? boolean
        | +-rw ipv6-multicast? boolean
        | ...
      +-rw isis-level-1-cfg
      | ...
      +-rw isis-level-2-cfg
      | ...
      +-rw overload
        | +-rw status?   boolean
        | +-rw timeout?  uint16
      +-rw interfaces
        +-rw interface* [name]
```

The diagram illustrates the structure of the configuration tree. It shows several levels of configuration nodes, each with its data type. Two specific nodes are highlighted with blue arrows pointing to labels: 'isis-level-1-cfg' and 'isis-level-2-cfg' are pointed to by arrows labeled 'Level specific configuration'; 'interface*' is pointed to by an arrow labeled 'Interface-specific configuration'.

Level specific configuration

Interface-specific configuration

Main trees : configuration (cont.)

```
+--rw isis-level-1-cfg
|   +-rw enabled?                                boolean
|   +-rw psnp-authentication?                     boolean
|   +-rw csnp-authentication?                     boolean
|   +-rw hello-authentication?                   boolean
|   +-rw authentication-key?                     string
|   +-rw authentication-type?                   enumeration
|   +-rw metric-type?                           enumeration
|   +-rw preference?                           uint8
|   +-rw external-preference?                 uint8
|   +-rw default-ipv4-unicast-metric?         isis-wide-metric
|   +-rw default-ipv6-unicast-metric?         isis-wide-metric
|   +-rw default-ipv4-multicast-metric?        isis-wide-metric
|   +-rw default-ipv6-multicast-metric?        isis-wide-metric
+-rw isis-level-2-cfg
|   +-rw enabled?                                boolean
|   +-rw psnp-authentication?                     boolean
|   +-rw csnp-authentication?                     boolean
|   +-rw hello-authentication?                   boolean
|   +-rw authentication-key?                     string
|   +-rw authentication-type?                   enumeration
|   +-rw metric-type?                           enumeration
|   +-rw preference?                           uint8
|   +-rw external-preference?                 uint8
|   +-rw default-ipv4-unicast-metric?         isis-wide-metric
|   +-rw default-ipv6-unicast-metric?         isis-wide-metric
|   +-rw default-ipv4-multicast-metric?        isis-wide-metric
|   +-rw default-ipv6-multicast-metric?        isis-wide-metric
```

Per level configuration

Main trees : configuration (cont.)

```
+--rw interfaces
    +-rw interface* [name]
        +-rw name                         leafref
        +-rw level?                       isis-level
        +-rw lsp-interval?                uint16
        +-rw passive?                    boolean
        +-rw csnp-interval?              uint16
        +-rw hello-authentication-type? enumeration
        +-rw hello-authentication-key?  string
        +-rw hello-interval?             uint16
        +-rw hello-multiplier?          uint16
        +-rw hello-padding?             boolean
        +-rw ipv4-unicast?              boolean
        +-rw ipv6-unicast?              boolean
        +-rw ipv4-multicast?            boolean
        +-rw ipv6-multicast?            boolean
        +-rw interface-type?            enumeration
        +-rw enabled?                   boolean
        +-rw tag*                        uint32
        +-rw level-1
            | +-rw hello-authentication-type?  enumeration
            | +-rw hello-authentication-key?  string
            | +-rw hello-interval?           uint16
            | +-rw hello-multiplier?         uint16
            | +-rw ipv4-unicast?            boolean
            | +-rw ipv6-unicast?            boolean
            | +-rw ipv4-multicast?          boolean
            | +-rw ipv6-multicast?          boolean
            | +-rw priority?               uint8
            | +-rw ipv4-unicast-metric?    isis-wide-metric
            | +-rw ipv6-unicast-metric?    isis-wide-metric
            | +-rw ipv4-multicast-metric?  isis-wide-metric
            | +-rw ipv6-multicast-metric?  isis-wide-metric
            | +-rw passive?                boolean
        +-rw level-2
            ...
        ...
```

Per interface configuration

Main trees : operational states

```
augment
  /rt:routing-state/rt:routing-instance/rt:routing-protocols/rt:routing-protocol:
    +-ro isis-state
      +-ro adjacencies
        |  +-ro adjacency* [interface] ← Operational adjacencies
        |  +-ro interface      string
        |  +-ro level?        isis-level
        |  +-ro state?        enumeration
      +-ro spf-log
        |  +-ro event* [id] ← Computation information
          +-ro id            uint32
          +-ro spf-type?     enumeration
          +-ro level?        isis-level
          +-ro spf-delay?    uint32
          +-ro schedule-timestamp? yang:timestamp
          +-ro start-timestamp? yang:timestamp
          +-ro end-timestamp? yang:timestamp
          +-ro trigger-lsp* [lsp]
            +-ro lsp           isis-lsp-id
            +-ro sequence?    uint32
      +-ro lsp-log
        |  +-ro event* [id] ← LSP events information
          +-ro id            uint32
          +-ro level?        isis-level
          +-ro lsp
            |  +-ro lsp?      isis-lsp-id
            |  +-ro sequence? uint32
            +-ro received-timestamp? yang:timestamp
      +-ro database
        |  +-ro level-1
        |  ...
        |  +-ro level-2
        |  ...
    +-ro hostnames
      +-ro hostname* [system-id] ← LSDB information
        +-ro system-id      isis-system-id
        +-ro hostname?     string
      
```

Main trees : operational states

```
augment /rt:routing-state/rt:ribs/rt:rib/rt:routes/rt:route:  
+--ro metric?          uint32  
+--ro tag*              uint32  
+--ro route-type?      enumeration
```

```
augment /rt:active-route/rt:output/rt:route:  
+--ro metric?          uint32  
+--ro tag*              uint32  
+--ro route-type?      enumeration
```

Add ISIS specific
informations to
routes

Main trees : RPC operations

```
rpcs:  
  +--x clear-isis-adjacency  
    |  +-ro input  
    |  +-ro routing-instance-name  
    |  +-ro routing-protocol-instance-name  
    |  +-ro isis-level?  
    |  +-ro interface?  
  +--x clear-isis-database  
    +-ro input  
      +-ro routing-instance-name  
      +-ro routing-protocol-instance-name  
      +-ro isis-level?  
  
                                rt:routing-instance-state-ref  
                                isis-instance-state-ref  
                                isis-level  
                                string  
  
                                rt:routing-instance-state-ref  
                                isis-instance-state-ref  
                                isis-level
```

Clear all or some ISIS adjacencies

Clear all or some LSDBs

Security considerations section addresses security issue introduced by such RPC (Access control required)

Main trees : notifications

```
notifications:  
  +---n isis-adjacency-updown  
    +-ro interface?          string  
    +-ro neighbor?           string  
    +-ro neighbor-system-id? isis-system-id  
    +-ro isis-level?         isis-level  
    +-ro state?              enumeration  
    +-ro reason?             string
```

A notification can be used to receive adjacency state change events

Feedbacks

- YANG structure/language sounds now better thanks to Netmod WG & YANG doctor
 - Version -01 is now fixing YANG specific part
 - RPC reply attached in APPENDIX
- ISIS specifics :
 - Discussion of naming of containers for LSDB model : use TLVxxx or more explicit names
 - Current version uses explicit names but points to TLV reference in description

What's next ?

- Look at feedback on the current proposed model from ISIS point of view :
 - What are the missing informations ?
 - Is the model organization OK for implementors ?
 - Need to agree on defaults, possibly feature specific containers, conditional leafs ...
- Looking at co-authors ...
- Could we consider this work as WG item ?