ISIS Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 26, 2014

S. Litkowski
Orange

June 24, 2014

Yang Data Model for ISIS protocol
draft-litkowski-isis-yang-isis-cfg-00

Abstract

This document defines a YANG data model that can be used to configure and manage ISIS protocol.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

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 December 26, 2014.

Copyright Notice

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

This document defines a YANG data model for ISIS routing protocol.

The data model covers configuration of an ISIS routing protocol instance as well as operational states.

1.1. Tree diagram

A simplified graphical representation of the data model is presented in Section 2. The meaning if the symbols in these diagrams is as follows:

- Brackets "[" and "]" enclose list keys.

- Symbols after data node names: "?" means an optional node, "!" means a presence container, and "*" denotes a list and leaf-list.

- Abbreviations before data node names: "rw" means configuration data (read-write), and "ro" means state data (read-only).

- Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").

- Ellipsis ("...") stands for contents of subtrees that are not shown.

2. Design of the data model

The ISIS Yang module is divided in two main containers :

- isis-cfg : that contains writable configuration objects.

- isis-state : that contains read-only states.

The container isis-cfg and isis-state are augmenting the "routing-protocol" lists in ietf-routing module with specific ISIS parameters.

The figure below describe the overall structure of the isis Yang module :

```
module: isis
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

augment /rt:routing/rt:routing-instance/rt:routing-protocols/rt:routing-protoco
   +--rw isis-cfg
      +--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 isis-level-1-cfg
      |  ...
      +--rw isis-level-2-cfg
```



```
      |  ...
      +--rw overload
      |  +--rw status?    boolean
      |  +--rw timeout?   uint16
      +--rw interfaces
         +--rw interface* [name]
            ...

augment /rt:routing-state/rt:routing-instance/rt:routing-protocols/rt:routing-p
   +--ro isis-state
      +--ro adjacencies
      |  +--ro adjacency* [interface]
      |     +--ro interface    string
      |     +--ro level?       isis-level
      |     +--ro state?       enumeration
      +--ro spf-log
      |  +--ro event* [id]
      |     +--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]
```

```
|     +--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]
      +--ro system-id    isis-system-id
      +--ro hostname?    string
```


2.1. ISIS configuration

The ISIS configuration container is divided in :

- Global parameters.
- Level specific parameters (see Section 2.3).
- Per interface configuration (see Section 2.4).

2.2. Multitopology parameters

The multitopology section is used to enable support of MT extensions for specific address families.

A boolean is associated with each topology type, defining if the topology is enabled or not.

```
      +--rw isis-multi-topology-cfg
      |  +--rw ipv4-unicast?     boolean
      |  +--rw ipv6-unicast?     boolean
      |  +--rw ipv4-multicast?   boolean
```

```
        |  +--rw ipv6-multicast?   boolean
```

2.3. Per level parameters

The level parameter section of the ISIS instance describes the global parameters for level 1 and 2 including authentication, protocol preferences, default metrics ...

Level specific parameters override parameters contained directly under isis-cfg container.


```
        +--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
```

2.4. Per interface parameters

The per-interface section of the ISIS instance describes the interface specific parameters.

Each interface has interface-specific parameters and level-specific parameters. A level-specific parameter always override interface-specific parameter and an interface-specific parameter always override an isis global parameter (defined in isis-cfg).

```
      +--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
               +--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
```

2.5. Operational states

isis-state container provides operational states for ISIS. This container is divided in multiple components :

- adjacencies : provides state information about current ISIS adjacencies.

- spf-log : provides information about SPF events on the node.

- lsp-log : provides information about LSP events on the node (reception of an LSP or modification of local LSP).

- database : provides details on current LSDB.

- hostname : provides information about system-id to hostname mappings.

3. RPC operations

The "ietf-isis" module defines two RPC operations :

- clear-isis-database : reset the content of a particular ISIS database and restart database synchronization with the neighbors.

- clear-isis-adjacency : restart a particular set of ISIS adjacencies.

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

4. Notifications

The "ietf-isis" module a new notification "isis-adjacency-updown". This notification is triggered when an ISIS adjacency moves to Up or Down state.

This notification provides details on the affected adjacency and its new state.

```
   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
```

5. Interaction with other YANG modules

The "isis-cfg" container augments the "/rt:routing/rt:routing-instance/rt:routing-protocols/routing-protocol" container of the ietf-routing module by defining ISIS specific parameters.

The "isis-state" container augments the "/rt:routing-state/rt:routing-instance/rt:routing-protocols/routing-protocol" container of the ietf-routing module by defining ISIS specific operational states.

Some ISIS specific routes attributes are added to route objects of the ietf-routing module by augmenting "/rt:routing-state/rt:ribs/rt:rib/rt:routes/rt:route" and "/rt:active-route/rt:output/rt:route".

6. Yang module

<CODE BEGINS> file "ietf-isis@2014-06-20.yang"

```
module ietf-isis {
    namespace "urn:ietf:params:xml:ns:yang:ietf-isis";

    prefix isis;

    import ietf-routing {
        prefix "rt";
    }

    import ietf-inet-types {
        prefix inet;
    }

    import ietf-yang-types {
        prefix yang;
    }
```

```
    organization
     "IETF ISIS Working Group";
```



```
    contact
        "WG List:   <mailto:isis-wg@ietf.org>

        Editor:     Stephane Litkowski
                    <mailto:stephane.litkowski@orange.com>";

    description
        "The YANG module defines a generic configuration model for ISIS
        common across all of the vendor implementations.";


    revision 2014-06-20 {
        description "
        * isis-op renamed to isis-state.
        * Multiple instances under isis are removed.
        * interface-cfg grouping removed and content
          is directly included in container isis.
        * TLVxx renamed with human-readable name in isis-database.
          TLV reference are putted in description.
        * Reference to core routing module were fixed.
        * Namespace fixed.
        * Add simple-iso-address type.
        * area-id and system-id in isis container are merged to
          nsap-address.
        * Add isis-system-id type.
        * Add isis-lsp-id type.
        * Add remaining-lifetime leaf in isis-database.
        * Add TLV2 (is-neighbor) in isis-database.
        * Renamed some container name for consistency
          reason ('isis-' prefixed).
        * Add new identities isis-cfg and isis-state.
        * Add descriptions.
        * Add notification isis-adjacency-updown.
        * Add RPC clear-isis-adjacency and clear-isis-database.
        ";
        reference "draft-litkowski-isis-yang-isis-00";
    }

    revision 2014-06-11 {
```

```
        description "Initial revision.";
        reference "draft-litkowski-netmod-isis-cfg-00";
    }
    identity isis {
        base rt:routing-protocol;
        description "Identity for the ISIS routing protocol.";
    }

    identity isis-cfg {
```



```
        description "Identity for the ISIS routing protocol
         configuration.";
    }

    identity isis-state {
        description "Identity for the ISIS routing protocol
         operational states.";
    }

    identity isis-adjacency-updown {
        description "Identity for the ISIS routing protocol
          adjacency state.";
    }

    identity clear-isis-database {
        description "Identity for the ISIS routing protocol
          database reset action.";
    }

    identity clear-isis-adjacency {
        description "Identity for the ISIS routing protocol
          adjacency reset action.";
    }

    typedef isis-instance-state-ref {
        type leafref {
            path "/rt:routing-state/rt:routing-instance/"
            +"rt:routing-protocols/rt:routing-protocol/rt:name";
        }
        description
         "This type is used for leafs that reference state data of
          an ISIS protocol instance.";
```

```
    }


    typedef isis-level {
        type enumeration {
            enum "level-1" {
                description
                 "This enum describes L1 only capability.";
            }
            enum "level-2" {
                description
                 "This enum describes L2 only capability.";
            }
            enum "level-1-2" {
                description
                 "This enum describes both level capability.";
```



```
            }
        }
        description
         "This type defines ISIS level of an object.";

    }

    typedef isis-lsp-id {
        type string {
            pattern
             "[0-9A-Fa-f]{4}\.[0-9A-Fa-f]{4}\.[0-9A-Fa-f]"
             +"{4}.[0-9][0-9]-[0-9][0-9]";
        }
        description
         "This type defines isis LSP ID using pattern,
          system id looks like : 0143.0438.AeF0.02-01";
    }
    typedef simple-iso-address {
        type string {
            pattern "[0-9A-Fa-f]{2}\.([0-9A-Fa-f]{4}\.){0,3}"
            +"[0-9A-Fa-f]{4}\.[0-9A-Fa-f]{4}\.[0-9A-Fa-f]{4}."
            +"[0-9][0-9]";
        }
        description
         "This type defines simple iso address format,
```

```
          it looks like : area_id.systemid.nsel
          The area ID is at least 1 byte of AFI, and is up to 13 bytes.";
    }

    typedef isis-system-id {
        type string {
            pattern "[0-9A-Fa-f]{4}\.[0-9A-Fa-f]{4}\.[0-9A-Fa-f]{4}.00";
        }
        description
         "This type defines isis system id using pattern,
          system id looks like : 0143.0438.AeF0.00";
    }

    typedef isis-wide-metric {
        type uint32 {
            range "0 .. 16777215";
        }
        description
         "This type defines wide style format
          of ISIS metric.";
    }

    typedef isis-std-metric {
```



```
        type uint8 {
            range "0 .. 63";
        }
        description
         "This type defines old style format
          of ISIS metric.";
    }

    grouping isis-route-content {
        description
         "This group add isis-specific route properties.";
        leaf metric {
            type uint32;
            description
             "This leaf describes isis metric of a route.";
        }
        leaf-list tag {
            type uint32;
```

```
            description
             "This leaf describes list of tags associated
              with the route.";
        }
        leaf route-type {
            type enumeration {
                enum l2-up-internal {
                    description "Level 2 internal route
                     and not leaked to a lower level";
                }
                enum l1-up-internal {
                    description "Level 1 internal route
                     and not leaked to a lower level";
                }
                enum l2-up-external {
                    description "Level 2 external route
                     and not leaked to a lower level";
                }
                enum l1-up-external {
                    description "Level 1 external route
                     and not leaked to a lower level";
                }
                enum l2-down-internal {
                    description "Level 2 internal route
                     and leaked to a lower level";
                }
                enum l1-down-internal {
                    description "Level 1 internal route
                     and leaked to a lower level";
                }
```



```
                enum l2-down-external {
                    description "Level 2 external route
                     and leaked to a lower level";
                }
                enum l1-down-external {
                    description "Level 1 external route
                     and leaked to a lower level";
                }
            }
            description
             "This leaf describes the type of isis route.";
```

```
        }
    }

    augment "/rt:routing-state/rt:ribs/rt:rib/rt:routes/rt:route" {
        when "rt:source-protocol = 'isis:isis'" {
            description "ISIS-specific route attributes.";
        }
        uses isis-route-content;
        description
         "This augments route object in RIB with ISIS-specific
          attributes.";
      }

    augment "/rt:active-route/rt:output/rt:route"
       {
        uses isis-route-content;
        description "ISIS-specific route attributes.";
      }


    grouping isis-prefix-ipv4-std {
        description
         "This group defines attributes of an
          IPv4 standard prefix.";
        leaf up-down {
            type boolean;
            description
             "This leaf expresses the value of up/down bit.";
        }
        leaf i-e {
            type boolean;
            description
             "This leaf expresses the value of I/E bit.";
        }
        leaf ip-prefix {
            type inet:ipv4-address;
            description
```



```
             "This leaf describes the IPv4 prefix";
        }
        leaf prefix-len {
            type uint8;
```

```
        description
         "This leaf describes the IPv4 prefix len in bits";
      }
      leaf default-metric {
        type isis-std-metric;
        description
         "This leaf describes the isis default metric value";
      }
      container delay-metric {
        leaf metric {
            type isis-std-metric;
            description
             "This leaf describes the isis delay metric value";
        }
        leaf supported {
            type boolean;
            default "false";
            description
             "This leaf describes if the metric is supported.";
        }

        description
         "This container defines the ISIS delay metric.";
      }
      container expense-metric {
        leaf metric {
            type isis-std-metric;
            description
             "This leaf describes the isis delay metric value";
        }
        leaf supported {
            type boolean;
            default "false";
            description
             "This leaf describes if the metric is supported.";
        }
        description
         "This container defines the ISIS expense metric.";
      }
      container error-metric {
        leaf metric {
            type isis-std-metric;
            description
             "This leaf describes the isis delay metric value";
```

```
            }
            leaf supported {
                type boolean;
                default "false";
                description
                 "This leaf describes if the metric is supported.";
            }

            description
             "This container defines the ISIS error metric.";
        }
    }

    grouping isis-prefix-ipv4-extended {
        description
         "This group defines attributes of an
          IPv4 extended prefix.";
        leaf up-down {
            type boolean;
            description
             "This leaf expresses the value of up/down bit.";
        }
        leaf ip-prefix {
            type inet:ipv4-address;
            description
             "This leaf describes the IPv4 prefix";
        }
        leaf prefix-len {
            type uint8;
            description
             "This leaf describes the IPv4 prefix len in bits";
        }

        leaf metric {
            type isis-wide-metric;
            description
             "This leaf describes the isis metric value";
        }
        leaf-list tag {
            type uint32;
            description
             "This leaf describes a list of tags associated with
             the prefix.";
        }
    }

    grouping isis-prefix-ipv6-extended {
```

```
        description
```



```
         "This group defines attributes of an
          IPv6 prefix.";
        leaf up-down {
            type boolean;
            description
             "This leaf expresses the value of up/down bit.";
        }
        leaf ip-prefix {
            type inet:ipv6-address;
            description
             "This leaf describes the IPv6 prefix";
        }
        leaf prefix-len {
            type uint8;
            description
             "This leaf describes the IPv4 prefix len in bits";
        }

        leaf metric {
            type isis-wide-metric;
            description
             "This leaf describes the isis metric value";
        }
        leaf-list tag {
            type uint32;
            description
             "This leaf describes a list of tags associated with
             the prefix.";
        }
    }

    grouping isis-neighbor-extended {
        description
         "This group defines attributes of an
          ISIS extended neighbor.";
        leaf neighbor-id {
            type isis-system-id;
            description
             "This leaf describes the system-id of the neighbor.";
        }
```

```
        leaf metric {
            type isis-wide-metric;
            description
             "This leaf describes the isis metric value";
        }
    }

    grouping isis-neighbor {
```



```
        description
         "This group defines attributes of an
          ISIS standard neighbor.";
        leaf neighbor-id {
            type isis-system-id;
            description
             "This leaf describes the system-id of the neighbor.";
        }
        leaf i-e {
            type boolean;
            description
             "This leaf expresses the value of I/E bit.";
        }
        leaf default-metric {
            type isis-std-metric;
            description
             "This leaf describes the isis default metric value";
        }
        container delay-metric {
            leaf metric {
                type isis-std-metric;
                description
                 "This leaf describes the isis delay metric value";
            }
            leaf supported {
                type boolean;
                default "false";
                description
                 "This leaf describes if the metric is supported.";
            }
            description
             "This container defines the ISIS delay metric.";
        }
```

```
        container expense-metric {
            leaf metric {
                type isis-std-metric;
                description
                 "This leaf describes the isis delay metric value";
            }
            leaf supported {
                type boolean;
                default "false";
                description
                 "This leaf describes if the metric is supported.";
            }
            description
             "This container defines the ISIS expense metric.";
        }
```



```
        container error-metric {
            leaf metric {
                type isis-std-metric;
                description
                 "This leaf describes the isis delay metric value";
            }
            leaf supported {
                type boolean;
                default "false";
                description
                 "This leaf describes if the metric is supported.";
            }
            description
             "This container defines the ISIS error metric.";
        }
    }

    grouping isis-database {
        description
         "This group defines attributes of an
          ISIS database (Link State DB).";
        leaf lsp-id {
            type isis-lsp-id;
            description
             "This leaf describes the LSP ID of the LSP.";
        }
```

```
        leaf checksum {
            type uint16;
            description
             "This leaf describes the checksum of the LSP.";
        }
        leaf remaining-lifetime {
            type uint16;
            units "seconds";
            description
             "This leaf describes the remaining lifetime
              in seconds before the LSP expiration.";
        }
        leaf sequence {
            type uint32;
            description
             "This leaf describes the sequence number of the LSP.";
        }
        leaf attributes {
            type bits {
                bit PARTITIONNED {
                    description
                     "If set, the originator supports partition
```



```
                      repair.";
                }
                bit ATTACHED-ERROR {
                    description
                     "If set, the originator is attached to
                      another area using the refered metric.";
                }
                bit ATTACHED-EXPENSE {
                    description
                     "If set, the originator is attached to
                      another area using the refered metric.";
                }
                bit ATTACHED-DELAY {
                    description
                     "If set, the originator is attached to
                      another area using the refered metric.";
                }
                bit ATTACHED-DEFAULT {
                    description
```

```
                "If set, the originator is attached to
                 another area using the refered metric.";
            }
            bit OVERLOAD {
                description
                 "If set, the originator is overloaded,
                  and must be avoided in path calculation.";
            }
        }
        description
         "This leaf describes attributes of the LSP.";
    }

    container is-neighbor {
        list neighbor {
            key "neighbor-id";
            uses isis-neighbor-extended;
            description
             "List of neighbors.";
        }
        description
         "This leaf describes list of ISIS extended neighbors.
         ISIS reference is TLV 2.";
    }

    container authentication {
        leaf authentication-type {
            type uint8;
            description
```



```
                "This leaf describes the authentication type
                 to be used.";
            }
            leaf authentication-key {
                type string;
                description
                 "This leaf describes the authentication key
                  to be used.";
            }
            description "This container describes authentication
            information of the node. ISIS reference is TLV 10.";
        }
```

```
        container extended-is-neighbor {
            list neighbor {
                key "neighbor-id";
                uses isis-neighbor-extended;
                description
                 "List of neighbors.";
            }
            description
             "This container describes list of ISIS extended neighbors.
             ISIS reference is TLV 22.";
        }

        container ipv4-internal-reachability {
            list prefixes {
                key "ip-prefix";
                uses isis-prefix-ipv4-std;
                description
                 "List of prefixes.";
            }
            description
             "This container describes list of ipv4 internal
             reachability information.
             ISIS reference is TLV 128.";
        }

        leaf-list protocol-supported {
            type uint8;
            description
             "This leaf describes the list of
             supported protocols.
             ISIS reference is TLV 129.";
        }

        container ipv4-external-reachability {
            list prefixes {
```



```
                key "ip-prefix";
                uses isis-prefix-ipv4-std;
                description
                 "List of prefixes.";
            }
```

```
            description
             "This container describes list of ipv4 external
             reachability information.
             ISIS reference is TLV 130.";
        }

        leaf-list ipv4-addresses {
            type inet:ipv4-address;
            description
                "This leaf describes the ipv4 addresses of the node.
                ISIS reference is TLV 132.";
        }

        leaf ipv4-te-routerid {

            type inet:ipv4-address;
            description
                "This leaf describes the IPv4 Traffic Engineering
                router ID of the node.
                ISIS reference is TLV 134.";
        }

        container extended-ipv4-reachability {

            list prefixes {
                key "ip-prefix";
                uses isis-prefix-ipv4-extended;
                description
                 "List of prefixes.";
            }
            description
             "This container describes list of ipv4 extended
             reachability information.
             ISIS reference is TLV 135.";
        }

        leaf dynamic-hostname {
            type string;

            description
             "This leaf describes the name of the node.
             ISIS reference is TLV 137.";
        }
```

```
leaf ipv6-te-routerid {
    type inet:ipv6-address;
    description
     "This leaf describes the IPv6 Traffic Engineering
      router ID of the node.
      ISIS reference is TLV 140.";
}


container mt-is-neighbor {
    list neighbor {
        key "neighbor-id";
        leaf MT-ID {
            type uint16 {
                range "0 .. 4095";
            }
            description
             "This leaf defines the identifier
              of a topology.";
        }
        uses isis-neighbor-extended;
        description
         "List of neighbors.";
    }
    description
     "This container describes list of ISIS multi-topology
      neighbors.
      ISIS reference is TLV 223.";
}

container mt-entries {
    list topology {
        key "MT-ID";

        leaf MT-ID {
            type uint16 {
                range "0 .. 4095";
            }
            description
             "This leaf defines the identifier
              of a topology.";
        }

        leaf attributes {
            type bits {
                bit OVERLOAD {
                    description
                     "If set, the originator is overloaded,
```

```
                            and must be avoided in path
                            calculation.";
                        }
                        bit ATTACHED {
                            description
                             "If set, the originator is attached to
                              another area using the refered metric.";
                        }
                    }
                    description
                     "This leaf describes attributes of the LSP
                      for the associated topology.";
                }
                description
                 "List of topologies supported.";
            }
            description
             "This container describes the topology supported.
             ISIS reference is TLV 229.";
        }

        leaf-list ipv6-addresses {
            type inet:ipv6-address;
            description
             "This leaf describes the ipv6 interface
             addresses of the node.
             ISIS reference is TLV 232.";
        }




        container mt-extended-ipv4-reachability {
            list prefixes {
                key "ip-prefix";
                leaf MT-ID {
                    type uint16 {
                        range "0 .. 4095";
                    }
                    description
                     "This leaf defines the identifier
                      of a topology.";
```

```
                }
                uses isis-prefix-ipv4-extended;
                description
                 "List of prefixes.";

            }
```



```
            description
             "This container describes list of ipv4
             reachability information in multi-topology
             environment.
             ISIS reference is TLV 235.";
        }

        container mt-ipv6-reachability {
            list prefixes {
                key "ip-prefix";
                leaf MT-ID {
                    type uint16 {
                        range "0 .. 4095";
                    }
                    description
                     "This leaf defines the identifier
                      of a topology.";
                }
                uses isis-prefix-ipv6-extended;
                description
                 "List of prefixes.";
            }
            description
             "This container describes list of ipv6
             reachability information in multi-topology
             environment.
             ISIS reference is TLV 237.";
        }

        container ipv6-reachability {
            list prefixes {
                key "ip-prefix";
                uses isis-prefix-ipv6-extended;
                description
                 "List of prefixes.";
```

```
            }
            description
             "This container describes list of ipv6
             reachability information.
             ISIS reference is TLV 236.";
        }

        container router-capabilities {
            leaf binary {
                type binary;
                description
                 "This leaf describes the capability of the node.
                 Format is binary according to the protocol encoding.";
```



```
            }
            description
             "This container describes the capabilities of the node.
             This container may be extended with detailed
             information.
             ISIS reference is TLV 242.";
        }
    }

    grouping isis-address-family-cfg {
        description
         "This group defines address-family-cfg
          global configuration for ISIS.";
        leaf ipv4-unicast {
            type boolean;
            description
             "This leaf defines if IPv4 unicast is activated.";
        }
        leaf ipv6-unicast {
            type boolean;
            description
             "This leaf defines if IPv6 unicast is activated.";
        }
        leaf ipv4-multicast {
            type boolean;
            description
             "This leaf defines if IPv4 multicast is activated.";
        }
```

```
        leaf ipv6-multicast {
            type boolean;
            description
             "This leaf defines if IPv6 multicast is activated.";
        }
    }

    grouping isis-interface-hello-cfg {
        description
         "This group defines hello interface
          parameters for ISIS.";
        leaf hello-authentication-type {
            type enumeration {
                enum    none {
                    description "No authentication used.";
                }
                enum    plaintext {
                    description "Plain text password used.";
                }
                enum    message-digest {
```



```
                    description "MD5 digest used.";
                }
            }
            description
             "This leaf describes the authentication type
             to be used in hello messages.";
        }
        leaf hello-authentication-key {
            type string;
            description
             "This leaf describes the authentication key
             to be used in hello messages.";
        }
        leaf hello-interval {
            type uint16;
            units "seconds";
            description
             "This leaf defines the interval of hello messages.";
        }
        leaf hello-multiplier {
            type uint16;
```

```
            description
             "This leaf defines the number of hello failed to be
              received before declaring the adjacency down.";
        }
    }

    grouping isis-interface-level-cfg {
        description
         "This group defines level specific
          configuration for ISIS interfaces.";
        uses isis-interface-hello-cfg;

        uses isis-address-family-cfg;

        leaf priority {
            type uint8 {
                range "0 .. 127";
            }

            description
             "This leaf describes the priority of the interface
             for DIS election.";
        }
        leaf ipv4-unicast-metric {
            type isis-wide-metric;
            description
             "This leaf describes the IPv4 unicast metric
```



```
             of the interface.";
        }
        leaf ipv6-unicast-metric {
            type isis-wide-metric;
            description
             "This leaf describes the IPv6 unicast metric
             of the interface.";
        }
        leaf ipv4-multicast-metric {
            type isis-wide-metric;
            description
             "This leaf describes the IPv4 multicast metric
             of the interface.";
        }
```

```
        leaf ipv6-multicast-metric {
            type isis-wide-metric;
            description
             "This leaf describes the IPv6 multicast metric
             of the interface.";
        }
        leaf passive {
            type boolean;
            default "false";
            description
             "This leaf defines if interface is in passive mode
             (ISIS not running, but network is advertised).";
        }
    }

    grouping isis-authentication-cfg {
        description
         "This group defines authentication
         configuration for ISIS.";
        leaf psnp-authentication {
            type boolean;
            default "true";
            description
             "This leaf describes if PSNP messages must be
             authenticated.";
        }
        leaf csnp-authentication {
            type boolean;
            default "true";
            description
             "This leaf describes if CSNP messages must be
             authenticated.";
        }
        leaf hello-authentication {
```



```
            type boolean;
            default "true";
            description
             "This leaf describes if HELLO messages must be
             authenticated.";
        }
        leaf authentication-key {
```

```
            type string;
            description
             "This leaf describes the authentication key
             to be used.";
        }
        leaf authentication-type {
            type enumeration {
                enum    none {
                    description "No authentication used.";
                }
                enum    plaintext {
                    description "Plain text password used.";
                }
                enum    message-digest {
                    description "MD5 digest used.";
                }
            }
            description
             "This leaf describes the authentication type
             to be used.";
        }
    }

    grouping isis-level-cfg {
        description
         "This group defines level specific
          global configuration for ISIS.";
        leaf enabled {
            type boolean;
            default "true";
            description
             "This leaf defines the status of the administrative
              status of the level (Active / not active).";
        }

        uses isis-authentication-cfg;

        leaf metric-type {
            type enumeration {
                enum wide-only {
                    description "Advertise new metric style only (RFC5305)";
```

```
                }
                enum old-only {
                    description "Advertise old metric style only (RFC1195)";
                }
                enum both {
                    description "Advertise both metric styles";
                }
            }
            description
             "This leaf describes the type of metric to be generated.
             Wide-only means only new metric style is generated, old-only
             means that only old style metric is generated, and both means
             that both are advertised.";
        }
        leaf preference {
            type uint8;
            description
             "This leaf defines the protocol preference.";
        }
        leaf external-preference {
            type uint8;
            description
             "This leaf defines the protocol preference for external routes.";
        }
        leaf default-ipv4-unicast-metric {
            type isis-wide-metric;
            description
             "This leaf defines the IPv4 unicast default metric.";
        }
        leaf default-ipv6-unicast-metric {
            type isis-wide-metric;
            description
             "This leaf defines the IPv6 unicast default metric.";
        }
        leaf default-ipv4-multicast-metric {
            type isis-wide-metric;
            description
             "This leaf defines the IPv4 multicast default metric.";
        }
        leaf default-ipv6-multicast-metric {
            type isis-wide-metric;
            description
             "This leaf defines the IPv6 multicast default metric.";
        }
    }

    augment "/rt:routing/rt:routing-instance/rt:routing-protocols/"
        +"rt:routing-protocol" {
```

```
        when "rt:type = 'isis:isis'" {
            description
             "This augment is only valid when routing protocol
              instance type is isis.";
        }
        description
         "This augments a routing protocol instance with ISIS
          specific parameters.";
        container isis-cfg {
            leaf isis-level {
                type isis-level;
                default "level-1-2";
                description
                 "This leaf describes the type of ISIS node.
                 A node can be level-1-only, level-2-only or level-1-2.
                 ";
            }
            leaf nsap-address {
                type simple-iso-address;
                description
                 "This leaf defines the NSAP address of the node
                  in a simple format. This parameter is mandatory.";
            }
            leaf ipv4-router-id {
                type inet:ipv4-address;
                description
                 "Router ID value that would be used in TLV134.";
            }
            leaf ipv6-router-id {
                type inet:ipv6-address;
                description
                 "Router ID value that would be used in TLV234.";
            }
            leaf reference-bandwidth {
                type uint32;
                units "bps";
                description
                 "This leaf defines the bandwidth for calculating metric.";
            }

            leaf lsp-mtu {
                type uint16;
                units "bytes";
```

```
            default 1492;
            description
             "This leaf describes the maximum size of a
              LSP PDU in bytes.";
          }
```



```
          leaf lsp-lifetime {
              type uint16;
              units "seconds";
              description
               "This leaf describes the lifetime of the router
                LSP in seconds.";
          }
          leaf lsp-refresh {
              type uint16;
              units "seconds";
              description
               "This leaf describes the refresh interval of the
                router LSP in seconds.";
          }

          uses isis-authentication-cfg;

          container isis-multi-topology-cfg {
              uses isis-address-family-cfg;
              description
               "This container describes activation of MT extensions
                for supporting new address families.";
          }

          container isis-level-1-cfg {
              uses isis-level-cfg;
              description
               "Defines configuration parameters of level 1.";
          }

          container isis-level-2-cfg {
              uses isis-level-cfg;
              description
               "Defines configuration parameters of level 2.";

          }
```

```
            container overload {
                leaf status {
                    type boolean;
                    description
                     "This leaf defines the overload status.";
                }
                leaf timeout {
                    type uint16;
                    units "seconds";
                    description
                     "This leaf defines the timeout in seconds
```



```
                           of the overload condition.";
                }
                description
                 "This leaf describes if the router is
                  set to overload state.";
            }

            container interfaces {
                list interface {
                    key "name";
                    leaf name {
                        type leafref {
                            path "/rt:routing/rt:routing-instance/"
                            +"rt:interfaces/rt:interface/rt:name";
                        }
                        description
                         "Reference to the interface within
                          the routing-instance.";
                    }
                    leaf level {
                        type isis-level;
                        default "level-1-2";
                        description
                         "This leaf defines the associated ISIS level
                          of the interface.";
                    }
                    leaf lsp-interval {
                        type uint16;
                        units "milliseconds";
```

```
                    description
                     "This leaf defines the interval between LSP
                      transmissions in msec";
                }
                leaf passive {
                    type boolean;
                    default "false";
                    description
                     "This leaf defines if interface is in passive mode
                      (ISIS not running, but network is advertised).";
                }
                leaf csnp-interval {
                    type uint16;
                    units "seconds";
                    description
                     "This leaf defines the interval of CSNP
                      messages.";
                }
```



```
                uses isis-interface-hello-cfg;

                leaf hello-padding {
                    type boolean;
                    description
                     "This leaf defines if ISIS Hellos would be
                      padded up to MTU size.";
                }

                uses isis-address-family-cfg;


                leaf interface-type {
                    type enumeration {
                        enum broadcast {
                            description "Broadcast interface type.
                            Would result in DIS election.";
                        }
                        enum point-to-point {
                            description "Point to point interface type.";
                        }
                    }
```

```
                    description
                     "This leaf defines the type of adjacency
                     to be established on the interface. This is
                     affecting the type of hello message that would
                     be used.";
                }

                leaf enabled {
                    type boolean;
                    default "true";
                    description
                     "This leaf describes the administrative status
                      of the ISIS interface.";
                }

                leaf-list tag {
                    type uint32;
                    description
                     "This leaf defines list of tags associated
                      with the interface.";
                }

                container level-1 {
                    uses isis-interface-level-cfg;
                    description
                     "This container defines the level 1 specific
```



```
                     configuration of the interface.";
                }

                container level-2 {
                    uses isis-interface-level-cfg;
                    description
                     "This container defines the level 2 specific
                     configuration of the interface.";
                }

                description
                 "List of ISIS interfaces.";
            }
            description
             "This container defines ISIS interface specific
```

```
                configuration objects.";
            }

            description
             "This container defines ISIS specific configuration
              objects.";
        }
    }

    augment "/rt:routing-state/rt:routing-instance/"
        +"rt:routing-protocols/rt:routing-protocol" {
        when "rt:type = 'isis:isis'" {
            description
             "This augment is only valid when routing protocol
              instance type is isis.";
        }
        description
         "This augments routing protocol instance states with ISIS
          specific parameters.";
        container isis-state {
            config false;
            container adjacencies {

                list adjacency {
                    key interface;

                    leaf interface {
                        type string;
                        description
                         "This leaf describes the name
                         of the interface.";
                    }
                    leaf level {
```



```
                        type uint8 {
                            range "1 .. 2";
                        }
                        description
                            "This leaf describes the associated
                            ISIS level of the interface.
                            The value of the level can only be 1
                            or 2.";
```

```
                }
                leaf state {
                    type enumeration {
                        enum "Up" {
                            description
                             "This state describes that
                              adjacency is established.";
                        }
                        enum "Down" {
                            description
                             "This state describes that
                              adjacency is NOT established.";
                        }
                        enum "Init" {
                            description
                             "This state describes that
                              adjacency is establishing.";
                        }
                    }
                    description
                     "This leaf describes the state of the
                     interface.";
                }
                description
                 "List of operational adjacencies.";
            }
            description
             "This container lists the adjacencies of
             the local node.";
        }
        container spf-log {
            list event {
                key id;

                leaf id {
                    type uint32;
                    description
                     "This leaf defines the event identifier.
                      This is a purely internal value.";
                }
```



```
                leaf spf-type {
```

```
                type enumeration {
                    enum full {
                        description
                         "Computation done is a Full SPF.";
                    }
                    enum incremental {
                        description
                         "Computation done is an
                          incremental SPF.";
                    }
                    enum route-only {
                        description
                         "Computation done is a
                          reachability computation
                          only.";
                    }
                }
                description
                 "This leaf describes the type of computation
                 used.";
            }
            leaf level {
                type uint8 {
                    range "1 .. 2";
                }
                description
                 "This leaf describes the level affected bytes
                 the computation.";
            }
            leaf spf-delay {
                type uint32;
                units "milliseconds";
                description
                 "This leaf describes the SPF delay that
                 was used for this event.";
            }
            leaf schedule-timestamp {
                type yang:timestamp;
                description
                 "This leaf describes the timestamp
                  when the computation was scheduled.";
            }
            leaf start-timestamp {
                type yang:timestamp;
                description
                 "This leaf describes the timestamp
                  when the computation was started.";
```

```
                    }
                    leaf end-timestamp {
                        type yang:timestamp;
                        description
                         "This leaf describes the timestamp
                          when the computation was ended.";
                    }
                    list trigger-lsp {
                        key "lsp";
                        leaf lsp {
                            type isis-lsp-id;
                            description
                             "This leaf describes the LSPID
                              of the LSP.";
                        }
                        leaf sequence {
                            type uint32;
                            description
                             "This leaf describes the sequence
                              number of the LSP.";
                        }
                        description
                         "This leaf describes list of LSPs
                          that triggered the computation.";
                    }
                    description
                     "List of computation events.";
                }

                description
                 "This container lists the SPF computation events.";
            }
            container lsp-log {
                list event {
                    key id;

                    leaf id {
                        type uint32;
                        description
                         "This leaf defines the event identifier.
                          This is a purely internal value.";
                    }
                    leaf level {
                        type uint8 {
                            range "1 .. 2";
                        }
```

```
                    description
                     "This leaf describes the level affected bytes
```



```
                     the computation.";
                }
                container lsp {
                    leaf lsp {

                        type isis-lsp-id;
                        description
                         "This leaf describes the LSPID
                          of the LSP.";
                    }
                    leaf sequence {
                        type uint32;
                        description
                         "This leaf describes the sequence
                          number of the LSP.";
                    }
                    description
                     "This container describes the received LSP
                     , in case of local LSP update the local
                     LSP ID is referenced.";
                }

                leaf received-timestamp {
                    type yang:timestamp;

                    description
                     "This leaf describes the timestamp
                     when the LSP was received. In case of
                     local LSP update, the timestamp refers
                     to the local LSP update time.";
                }

                description
                 "List of LSP events.";
            }

            description
             "This container lists the LSP reception events.
             Local LSP modification are also contained in the
```

```
                list.";
            }
            container database {
                container level-1 {
                    list lsp {
                        key lsp-id;

                        uses isis-database;
                        description
```



```
                         "List of LSPs in LSDB.";
                    }

                    description
                     "This container describes the list of LSPs
                     in the level-1 database.";
                }
                container level-2 {
                    list lsp {
                        key lsp-id;

                        uses isis-database;
                        description
                         "List of LSPs in LSDB.";
                    }

                    description
                     "This container describes the list of LSPs
                     in the level-2 database.";
                }
                description
                 "This container describes ISIS Link State
                  databases.";
            }
            container hostnames {

                list hostname {
                    key system-id;
                    leaf system-id {
                        type isis-system-id;
                        description
                         "This leaf describes the system-id
```

```
                     associated with the hostname.";
                }
                leaf hostname {

                    type string;
                    description
                     "This leaf describes the hostname
                      associated with the system ID.";
                }
                description
                 "List of system-id/hostname associations";
            }

            description
             "This container describes the list
             of binding between system-id and
```



```
                 hostnames.";
            }

            description
             "This container defines various ISIS states objects.";
        }
    }

    /* RPC methods */

    rpc clear-isis-adjacency {
        description
         "This RPC request clears a particular
          set of ISIS adjacencies. If the operation
          fails for ISIS internal reason, then
          error-tag and error-app-tag should be set
          to a meaningful value.";
        input {
            leaf routing-instance-name {
                type rt:routing-instance-state-ref;
                mandatory "true";
                description
                 "Name of the routing instance whose ISIS
                  information is being queried.
```

```
                If the routing instance with name equal to the
                value of this parameter doesn't exist, then this
                operation SHALL fail with error-tag 'data-missing'
                and error-app-tag 'routing-instance-not-found'.";


            }
            leaf routing-protocol-instance-name {
                type isis-instance-state-ref;
                mandatory "true";
                description
                 "Name of the ISIS protocol instance whose ISIS
                  information is being queried.

                  If the ISIS instance with name equal to the
                  value of this parameter doesn't exist, then this
                  operation SHALL fail with error-tag 'data-missing'
                  and error-app-tag
                  'routing-protocol-instance-not-found'.";
            }
            leaf isis-level {
                type isis-level;
                description
```



```
                 "ISIS level of the adjacency to be cleared.
                 If ISIS level is level-1-2, both level 1 and level 2
                 adjacencies would be cleared.

                 If the value provided is different from the one
                 authorized in the enum type, then this
                  operation SHALL fail with error-tag 'data-missing'
                  and error-app-tag
                  'bad-isis-level'.
                 ";
            }
            leaf interface {
                type string;
                description
                 "Name of the ISIS interface.

                  If the ISIS interface with name equal to the
                  value of this parameter doesn't exist, then this
```

```
                operation SHALL fail with error-tag 'data-missing'
                and error-app-tag
                'isis-interface-not-found'.";
            }
        }
    }

    rpc clear-isis-database {
        description
         "This RPC request clears a particular
          ISIS database. If the operation
          fails for ISIS internal reason, then
          error-tag and error-app-tag should be set
          to a meaningful value.";
        input {
            leaf routing-instance-name {
                type rt:routing-instance-state-ref;
                mandatory "true";
                description
                 "Name of the routing instance whose ISIS
                  information is being queried.

                  If the routing instance with name equal to the
                  value of this parameter doesn't exist, then this
                  operation SHALL fail with error-tag 'data-missing'
                  and error-app-tag 'routing-instance-not-found'.";


            }
            leaf routing-protocol-instance-name {
```



```
                type isis-instance-state-ref;
                mandatory "true";
                description
                 "Name of the ISIS protocol instance whose ISIS
                  information is being queried.

                  If the ISIS instance with name equal to the
                  value of this parameter doesn't exist, then this
                  operation SHALL fail with error-tag 'data-missing'
                  and error-app-tag
                  'routing-protocol-instance-not-found'.";
```

```
            }
            leaf isis-level {
                type isis-level;
                description
                 "ISIS level of the adjacency to be cleared.
                 If ISIS level is level-1-2, both level 1 and level 2
                 adjacencies would be cleared.

                 If the value provided is different from the one
                 authorized in the enum type, then this
                  operation SHALL fail with error-tag 'data-missing'
                  and error-app-tag
                  'bad-isis-level'.
                 ";
            }
        }

    }



    /* Notifications */

    notification isis-adjacency-updown {
        leaf interface {
            type string;
            description
             "Describes the interface of the adjacency";
        }
        leaf neighbor {
            type string;
            description
             "Describes the name of the neighbor. If the
             name of the neighbor is not available, the
             field would be empty.";
        }
        leaf neighbor-system-id {
```



```
            type isis-system-id;
            description
             "Describes the system-id of the neighbor.";
        }
```

```
        leaf isis-level {
            type isis-level;
            description
             "Describes the ISIS level of the adjacency.";
        }
        leaf state {
            type enumeration {
                enum "Up" {
                    description
                     "This state describes that
                      adjacency is established.";
                }
                enum "Down" {
                    description
                     "This state describes that
                      adjacency is no more established.";
                }
            }
            description
             "This leaf describes the new state of the
              ISIS adjacency.";
        }
        leaf reason {
            type string;
            description
             "If the adjacency is going to DOWN,
             this leaf provides a reason for the adjacency
             going down. The reason is provided as a text.
             If the adjacency is going to UP, no reason is
             provided.";
        }
        description
          "This notification is sent when an ISIS adjacency
           moves to Up state or to Down state.";
    }

}

<CODE ENDS>
```

[I-D.ietf-netmod-routing-cfg]
Lhotka, L., "A YANG Data Model for Routing Management", draft-ietf-netmod-routing-cfg-15 (work in progress), May 2014.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

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

Appendix A. Example: NETCONF <get> Reply

This section gives an example of a reply to the NETCONF <get> request for a device that implements the data model defined in this document. The example is written in XML.

```
<?xml version="1.0" encoding="cp1252"?>
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <routing-state xmlns="urn:ietf:params:xml:ns:yang:ietf-routing">
        <routing-instance>
            <name>rtr1</name>
            <id>1</id>
            <router-id>192.0.2.1</router-id>
            <routing-protocols>
                <routing-protocol>
                    <name>ISIS1</name>
                    <type>isis</type>
                    <isis-state xmlns="urn:ietf:params:xml:ns:yang:ietf-isis">
                        <adjacencies>
                            <adjacency>
                                <interface>eth0</interface>
                                <level>2</level>
                                <state>Up</state>
                            </adjacency>
                        </adjacencies>
                        <spf-log>
                            <event>
```

```
                    <id>18979</id>
```



```
                    <spf-type>full</spf-type>
                    <level>2</level>
                    <spf-delay>150</spf-delay>
                    <schedule-timestamp>1403612245</schedule-timest
                    <start-timestamp>1403612399</start-timestamp>
                    <end-timestamp>1403612420</end-timestamp>
                    <trigger-lsp>
                        <lsp>0000.01245.1245.01-01</lsp>
                        <sequence>125458</sequence>
                    </trigger-lsp>
                </event>
            </spf-log>
            <lsp-log>
                <event>
                    <id>1245</id>
                    <level>2</level>
                    <lsp>
                        <lsp>0000.01245.1245.01-01</lsp>
                        <sequence>125458</sequence>
                    </lsp>
                    <received-timestamp></received-timestamp>
                </event>
            </lsp-log>
            <database>
                <level-2>
                    <lsp>
                        <lsp-id>0000.01245.1245.01-01</lsp-id>
                        <checksum>1245</checksum>
                        <remaining-lifetime>45</remaining-lifetime>
                        <sequence>125458</sequence>
                        <attributes />
                        <extended-is-neighbor>
                            <neighbor>
                                <neighbor-id>0000.01245.9999.00</ne
                                <metric>100</metric>
                            </neighbor>
                        </extended-is-neighbor>

                        <protocol-supported>204<protocol-supported>
```

```
                                    <ipv4-addresses>192.168.0.2</ipv4-addresses
                                    <extended-ipv4-reachability>
                                        <prefixes>
                                            <up-down>false</up-down>
                                            <ip-prefix>192.168.0.2</ip-prefix>
                                            <prefix-len>32</prefix-len>
                                            <metric>30490</metric>
                                            <tag>200</tag>
```



```
                                        </prefixes>
                                    </extended-ipv4-reachability>
                                    <dynamic-hostname>rtr1</dynamic-hostname>

                                </lsp>
                            </level-2>
                        </database>
                        <hostnames>
                            <hostname>
                                <system-id>0000.01245.9999.00</system-id>
                                <hostname>rtr1</hostname>
                            </hostname>
                        </hostnames>
                    </isis-state>
                </routing-protocol>
            </routing-protocols>
        </routing-instance>
        <ribs>
            <rib>
                <name>ipv4-master</name>
                <id>1</id>
                <address-family>v4ur:ipv4-unicast</address-family>
                <routes>
                    <route>
                        <id>124554657</id>
                        <outgoing-interface>eth0</outgoing-interface>
                        <source-protocol>isis</source-protocol>
                        <last-updated>2013-07-02T18:02:45+01:00</last-updated>
                        <destination-prefix xmlns="urn:ietf:params:xml:ns:yang:
                        10.0.0.0/24</destination-prefix>
                        <metric xmlns="urn:ietf:params:xml:ns:yang:ietf-isis">
                        750
                        </metric>
```

```
                    <route-type xmlns="urn:ietf:params:xml:ns:yang:ietf-isi

                </route>
            </routes>

        </rib>
    </ribs>

</routing-state>
<routing xmlns="urn:ietf:params:xml:ns:yang:ietf-routing">
    <routing-instance>
        <id>1</id>
        <router-id>192.0.2.1<router-id>
        <routing-protocols>
            <routing-protocol>
```



```
                    <name>ISIS1</name>
                    <type>isis</type>
                    <isis-cfg xmlns="urn:ietf:params:xml:ns:yang:ietf-isis">
                        <isis-level>level-2</isis-level>
                        <nsap-address>49.0002.0000.01245.9999.00</nsap-address>
                        <lsp-lifetime>65535</lsp-lifetime>
                        <lsp-refresh>65000</lsp-refresh>
                        <authentication-key>my_password</authentication-key>
                        <authentication-type>plaintext</authentication-type>
                        <isis-multi-topology-cfg>
                            <ipv4-unicast>true</ipv4-unicast>
                        </isis-multi-topology-cfg>

                        <isis-level-2-cfg>
                            <metric-type>wide-only</metric-type>
                            <default-ipv4-unicast-metric>11111111</default-ipv4
                        </isis-level-2-cfg>
                        <interfaces>
                            <interface>
                                <name>eth0</name>
                                <lsp-interval>100</lsp-interval>
                                <csnp-interval>10</csnp-interval>
                                <interface-type>point-to-point</interface-type>
                                <level-2>
                                    <ipv4-unicast-metric>200</ipv4-unicast-metr
                                </level-2>
```

```
                            </interface>
                            <interface>
                                <name>lo0</name>
                                <level-2>
                                    <ipv4-unicast-metric>1</ipv4-unicast-metric
                                </level-2>
                                <passive>true</passive>
                            </interface>
                        </interfaces>
                    </isis-cfg>
                </routing-protocol>
            </routing-protocols>
        </routing-instance>
    </routing>
</data>
```

Author's Address

Stephane Litkowski
Orange

Email: stephane.litkowski@orange.com