


```
    uses default-metric-global-cfg;

    container level-1 {
        uses default-metric-global-cfg;
        description "level-1 specific cfg";
    }
    container level-2 {
        uses default-metric-global-cfg;
        description "level-2 specific cfg";
    }
    description "Default metric global cfg.";
}

container afs {
    if-feature nlpid-control;
    list af {
        key af;
        leaf af {
            type identityref {
                base rt:address-family;
            }
            description
                "Address-family";
        }

        leaf enable {
            type boolean;
            description
                "Describes the activation state of the
                AF.";
        }
        description
            "This list permits activation
            of new address families.";
    }
    description
        "Container for address-families";
}

container preference {
    uses route-preference-global-cfg;
    description
        "This container defines the protocol preference.";
}

container overload {
```



```
        uses overload-global-cfg;
        description
        "This container describes if the router is
        set to overload state.";
    }

    container overload-max-metric {
        if-feature overload-max-metric;
        uses overload-max-metric-global-cfg;

        description
        "This container describes if the router is
        set to overload state using max-metric
        advertisement.";
    }

}

grouping isis-global-topologies-cfg {
    description
    "Per topology config.";

    container default-metric {
        uses default-metric-global-cfg;

        container level-1 {
            uses default-metric-global-cfg;
            description "level-1 specific cfg";
        }
        container level-2 {
            uses default-metric-global-cfg;
            description "level-2 specific cfg";
        }
        description "Default metric per
        topology cfg.";
    }

    uses isis-node-tag-cfg;
}

grouping isis-if-cfg {
    description
    "Grouping for interface cfg.";
```



```
leaf level-type {
    type level;
    default "level-all";
    description
        "This leaf defines the associated ISIS
        level of the interface.";
}
leaf lsp-pacing-interval {
    type uint16;
    units "milliseconds";
    default 33;
    description
        "This leaf defines the interval between
        LSP transmissions in milli-seconds";
}
leaf lsp-retransmit-interval {
    type uint16;
    units "seconds";
    description
        "This leaf defines the interval between
        retransmission of LSP";
}
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 {
        range "1..65535";
    }
    units "seconds";
    default 10;
    description
        "This leaf defines the interval of CSNP
        messages.";
}

container hello-padding {
    leaf enable {
        type boolean;
        default "true";
        description
            "Status of Hello-padding activation.
            By default, the implementation shall
```



```
        pad HELLOs.";
    }

    description
        "This container handles ISIS hello padding
        configuration.";
}

leaf mesh-group-enable {
    type mesh-group-state;
    description
        "Describes the mesh group state of
        the interface.";
}

leaf mesh-group {
    when "../mesh-group-enable = 'meshSet'" {
        description
            "Only valid when mesh-group-enable
            equals meshSet";
    }
    type uint8;
    description
        "Describes the mesh group ID of
        the interface.";
}

leaf interface-type {
    type 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.";
}

uses admin-control;

leaf-list tag {
    if-feature prefix-tag;

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



```
leaf-list tag64 {
    if-feature prefix-tag64;

    type uint64;
    description
        "This leaf defines list of 64bits tags
        associated with the interface.";
}

leaf node-flag {
    if-feature node-flag;
    type boolean;
    default false;
    description
        "Set prefix as a node
        representative prefix.";
}

container hello-authentication {
    uses hello-authentication-cfg;

    container level-1 {
        uses hello-authentication-cfg;
        description "level-1 specific cfg";
    }
    container level-2 {
        uses hello-authentication-cfg;
        description "level-2 specific cfg";
    }
    description "Authentication type
    to be used in hello messages.";
}

container hello-interval {
    uses hello-interval-cfg;

    container level-1 {
        uses hello-interval-cfg;
        description "level-1 specific cfg";
    }
    container level-2 {
        uses hello-interval-cfg;
        description "level-2 specific cfg";
    }
    description "Interval between
```



```
        hello messages.";
    }

    container hello-multiplier {
        uses hello-multiplier-cfg;

        container level-1 {
            uses hello-multiplier-cfg;
            description "level-1 specific cfg";
        }
        container level-2 {
            uses hello-multiplier-cfg;
            description "level-2 specific cfg";
        }
        description "Hello multiplier
        configuration.";
    }

    container priority {
        must '../interface-type = "broadcast"' {
            error-message
            "Priority only applies to broadcast
            interfaces.";
            description
            "Check for broadcast interface.";
        }
        uses priority-cfg;

        container level-1 {
            uses priority-cfg;
            description "level-1 specific cfg";
        }
        container level-2 {
            uses priority-cfg;
            description "level-2 specific cfg";
        }
        description "Priority for DIS election.";
    }

    container metric {
        uses metric-cfg;

        container level-1 {
            uses metric-cfg;
            description "level-1 specific cfg";
        }
        container level-2 {
            uses metric-cfg;
```



```
        description "level-2 specific cfg";
    }
    description "Metric configuration.";
}
```

```
container bfd {
    if-feature bfd;
    leaf enable {
        type boolean;
        default false;
        description "
            Enables BFD on the interface
        ";
    }
    description
        "BFD configuration.";
}
```

```
container afs {
    if-feature nlpid-control;
    list af {
        key af;

        leaf af {
            type identityref {
                base rt:address-family;
            }
            description
                "Address-family";
        }

        description
            "List of AFs.";
    }
    description
        "Container for address-families";
}
```

```
container mpls {
    container igp-ldp-sync {
        if-feature igp-ldp-sync;
        leaf enable {
            type boolean;
            description
                "Enable/disable IGP LDP sync.";
        }
    }
}
```



```
    }
    description
      "IGP-LDP sync configuration.";
  }
  description
    "Container for MPLS specific configuration
    for ISIS.";
}

grouping isis-if-topologies-cfg {
  description
    "ISIS interface topology cfg.";
  container metric {
    uses metric-cfg;

    container level-1 {
      uses metric-cfg;
      description "level-1 specific cfg";
    }
    container level-2 {
      uses metric-cfg;
      description "level-2 specific cfg";
    }
    description "Metric configuration.";
  }
}

/* */

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

augment "/if:interfaces/if:interface"
{
  leaf clns-mtu {
```



```
    type uint16;
    description
      "Defines CLNS MTU of the interface.";
  }
  description "ISO interface config.";
}

augment "/rt:routing/rt:control-plane-protocols/"
  +"rt:control-plane-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 {

    must "count(area-address) > 0" {
      error-message "At least one area-address
        must be configured.";
      description
        "Enforce configuration of at least one area.";
    }

    uses isis-global-cfg;

    container fast-reroute {
      if-feature fast-reroute;
      uses fast-reroute-global-cfg;
      description
        "IPFRR.";
    }
    container topologies {
      if-feature multi-topology;
      list topology {

        key "name";

        leaf enable {
          type boolean;
          description
            "Control enabling of topologies";
        }
      }
    }
  }
}
```



```
    leaf name {
      type leafref {
        path "../../../../../rt:ribs/rt:rib/rt:name";
      }

      description "RIB";
    }

    uses isis-global-topologies-cfg;
    container fast-reroute {
      if-feature fast-reroute;
      uses fast-reroute-global-cfg;
      description
        "IPFRR.";
    }

    description
      "List of topologies";
  }
  description
    "Container for multi-topology";
}

container interfaces {
  list interface {
    key "name";
    leaf name {
      type if:interface-ref;

      description
        "Reference to the interface within
        the routing-instance.";
    }
  }

  uses isis-if-cfg;
  container fast-reroute {
    if-feature fast-reroute;
    uses fast-reroute-if-cfg;
    description
      "IPFRR.";
  }
  container topologies {
    if-feature multi-topology;
    list topology {
      key name;

      leaf name {
        type leafref {
```



```
        path "../../../../../../../../../../../"+
        "rt:ribs/rt:rib/rt:name";
    }

    description
        "Name of RIB.";
    }
    container fast-reroute {
        if-feature fast-reroute;
        uses fast-reroute-if-cfg;
        description
            "IPFRR.";
    }
    uses isis-if-topologies-cfg;
    description
        "List of topologies.";
    }
    description
        "Container for multi-topology";
    }

    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:control-plane-protocols/rt:control-plane-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 {
    config false;
```



```
uses isis-global-cfg;
container fast-reroute {
    if-feature fast-reroute;
    uses fast-reroute-global-cfg;
    uses fast-reroute-global-state;
    description
        "IPFRR states.";
}

list topologies {
    key name;

    leaf name {
        type leafref {
            path "../.../.../..."
                +"rt:ribs/rt:rib/rt:name";
        }

        description
            "Name of RIB.";
    }
    container fast-route {
        if-feature fast-reroute;
        uses fast-reroute-global-cfg;
        uses fast-reroute-global-state;
        description
            "IPFRR states.";
    }
    description
        "List of topologies.";
}

container system-counters {
    list level {
        key level;

        leaf level {
            type level-number;
            description
                "This leaf describes the ISIS level.";
        }
        leaf corrupted-lsps {
            type uint32;
            description
                "Number of corrupted in-memory LSPs detected.
                LSPs received from the wire with a bad
                checksum are silently dropped and not counted.
                LSPs received from the wire with parse errors
```



```
        are counted by lsp-errors.";
    }
    leaf authentication-type-fails {
        type uint32;
        description
            "Number of authentication type mismatches.";
    }
    leaf authentication-fails {
        type uint32;
        description
            "Number of authentication key failures.";
    }
    leaf database-overload {
        type uint32;
        description
            "Number of times the database has become
            overloaded.";
    }
    leaf own-lsp-purge {
        type uint32;
        description
            "Number of times a zero-aged copy of the
            system's own LSP is received from some
            other node.";
    }
    leaf manual-address-drop-from-area {
        type uint32;
        description
            "Number of times a manual address
            has been dropped from the area.";
    }
    leaf max-sequence {
        type uint32;
        description
            "Number of times the system has attempted
            to exceed the maximum sequence number.";
    }
    leaf sequence-number-skipped {
        type uint32;
        description
            "Number of times a sequence number skip has
            occurred.";
    }
    leaf id-len-mismatch {
        type uint32;
        description
            "Number of times a PDU is received with
            a different value for ID field length
```



```
        from that of the receiving system.";
    }
    leaf partition-changes {
        type uint32;
        description
            "Number of partition changes detected.";
    }
    leaf lsp-errors {
        type uint32;
        description
            "Number of LSPs with errors we have
            received.";
    }
    leaf spf-runs {
        type uint32;
        description
            "Number of times we ran SPF at this level.";
    }
    description
        "List of supported levels.";
}
description
    "The container defines a list of counters
    for the IS.";
}

container interfaces {
    list interface {
        key interface;

        leaf interface {
            type string;
            description
                "This leaf describes the name
                of the interface.";
        }
        uses isis-if-cfg;
        container fast-reroute {
            if-feature fast-reroute;
            uses fast-reroute-if-cfg;
            description
                "IPFRR.";
        }
        uses adjacency-state;

        list topologies {
            key name;
```



```
leaf name {
  type leafref {
    path "../../../../../"
    +"../../rt:ribs/rt:rib/rt:name";
  }

  description
    "Name of RIB.";
}
uses isis-if-topologies-cfg;
container fast-reroute {
  if-feature fast-reroute;
  uses fast-reroute-if-cfg;
  description
    "IPFRR.";
}
uses adjacency-state;

description
  "List of topologies.";
}

container event-counters {
  leaf adjacency-changes {
    type uint32;
    description
      "The number of times an adjacency state
      change has occurred on this interface.";
  }
  leaf adjacency-number {
    type uint32;
    description
      "The number of adjacencies on this
      interface.";
  }
  leaf init-fails {
    type uint32;
    description
      "The number of times initialization of
      this interface has failed. This counts
      events such as PPP NCP failures.
      Failures to form an adjacency are counted
      by adjacency-rejects.";
  }
}
```



```
leaf adjacency-rejects {
    type uint32;
    description
        "The number of times an adjacency has been
        rejected on this interface.";
}
leaf id-len-mismatch {
    type uint32;
    description
        "The number of times an IS-IS PDU with an ID
        field length different from that for this
        system has been received on this interface.";
}
leaf max-area-addresses-mismatch {
    type uint32;
    description
        "The number of times an IS-IS PDU with
        according max area address field
        differs from that for
        this system has been received on this
        interface.";
}
leaf authentication-type-fails {
    type uint32;
    description
        "Number of authentication type mismatches.";
}
leaf authentication-fails {
    type uint32;
    description
        "Number of authentication key failures.";
}
leaf lan-dis-changes {
    type uint32;
    description
        "The number of times the DIS has changed
        on this interface at this level.
        If the interface type is point to point,
        the count is zero.";
}
description
    "Provides protocol event counters.";
}
container packet-counters {
    list level {
        key level;

        leaf level {
```



```
    type level-number;
    description
      "This leaf describes the ISIS level.";
  }

  container iih {
    leaf in {
      type uint32;
      description
        "Received PDUs.";
    }
    leaf out {
      type uint32;
      description
        "Sent PDUs.";
    }
    description
      "The number of IIH PDUs received/sent.";
  }

  container ish {
    leaf in {
      type uint32;
      description
        "Received PDUs.";
    }
    leaf out {
      type uint32;
      description
        "Sent PDUs.";
    }
    description
      "The number of ISH PDUs received/sent.";
  }

  container esh {
    leaf in {
      type uint32;
      description
        "Received PDUs.";
    }
    leaf out {
      type uint32;
      description
        "Sent PDUs.";
    }
    description
      "The number of ESH PDUs received/sent.";
  }

  container lsp {
```



```
    leaf in {
      type uint32;
      description
        "Received PDUs.";
    }
    leaf out {
      type uint32;
      description
        "Sent PDUs.";
    }
    description
      "The number of LSP PDUs received/sent.";
  }
  container psnp {
    leaf in {
      type uint32;
      description
        "Received PDUs.";
    }
    leaf out {
      type uint32;
      description
        "Sent PDUs.";
    }
    description
      "The number of PSNP PDUs received/sent.";
  }
  container csnp {
    leaf in {
      type uint32;
      description
        "Received PDUs.";
    }
    leaf out {
      type uint32;
      description
        "Sent PDUs.";
    }
    description
      "The number of CSNP PDUs received/sent.";
  }
  container unknown {
    leaf in {
      type uint32;
      description
        "Received PDUs.";
    }
    leaf out {
```



```
        type uint32;
        description
            "Sent PDUs.";
    }
    description
        "The number of unknown PDUs received/sent.";
    }
    description
        "List of supported levels.";
    }
    description
        "Provides packet counters per level.";
    }
    description
        "List of interfaces.";
    }
    description
        "The container defines operational parameters
        of interfaces.";
    }
}

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 level-number;
    description
      "This leaf describes the level affected by the
      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 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 level-number;
            description
                "This leaf describes the level affected by the
                the computation.";
        }
    }
    container lsp {
        leaf lsp {

            type 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.";
}

leaf change {
  type bits {
    bit refresh {
      position 0;
      description
        "Refresh LSP, nothing has changed.";
    }
    bit link-down {
      position 1;
      description
        "One or more links are down.";
    }
    bit link-up {
      position 2;
      description
        "One or more links are up.";
    }
    bit link-metric-change {
      position 3;
      description
        "One or more links experienced
        a metric change.";
    }
    bit link-other-change {
      position 4;
      description
        "One or more links experienced
        a change that does not affect state
        or metric.";
    }
    bit prefix-down {
      position 5;
      description
        "One or more links are down.";
    }
    bit prefix-up {
      position 6;
      description
```



```
        "One or more prefixes are up.";
    }
    bit prefix-metric-change {
        position 7;
        description
            "One or more prefixes experienced
            a metric change.";
    }
    bit prefix-other-change {
        position 8;
        description
            "One or more prefixes experienced
            a change that does not affect state
            or metric.";
    }
    bit other-change {
        position 9;
        description
            "One or more component changed that
            is not a prefix or link.";
    }
}
description
    "This leaf describes the type of change
    in the LSP.";
}

description
    "List of LSP events.";
}

description
    "This container lists the LSP reception events.
    Local LSP modification are also contained in the
    list.";
}
container database {
    list level-db {
        key level;

        leaf level {
            type level-number;
            description
                "Current level number";
        }
    }
    list lsp {
        key lsp-id;
```



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

    description
        "This container describes the list of LSPs
        in the level x database.";
    }

    description
        "This container describes ISIS Link State
        databases.";
    }
    container hostnames {

        list hostname {
            key system-id;
            leaf system-id {
                type 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-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-protocol-instance-name {
      type 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 level {
      type 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-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-protocol-instance-name {
      type 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 level {
      type 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 database-overload {
  uses notification-instance-hdr;
```



```
leaf overload {
  type enumeration {
    enum "off" {
      description
        "The system has left overload condition.";
    }
    enum "on" {
      description
        "The system is in overload condition.";
    }
  }
  description
    "Describes the new overload state of the instance.";
}
description
  "This notification is sent when an ISIS instance
  overload condition changes.";
}

notification lsp-too-large {
  uses notification-instance-hdr;
  uses notification-interface-hdr;

  leaf pdu-size {
    type uint32;
    description
      "Size of the PDU";
  }
  leaf lsp-id {
    type lsp-id;
    description
      "LSP ID.";
  }
  description
    "This notification is sent when we attempt
    to propagate an LSP that is larger than the
    dataLinkBlockSize for the circuit.
    The notification generation must be throttled
    with at least a 5 second gap.
    ";
}

notification corrupted-lsp-detected {
  uses notification-instance-hdr;
  leaf lsp-id {
    type lsp-id;
    description
```



```
        "LSP ID.";
    }
    description
        "This notification is sent when we find
         that an LSP that was stored in memory has
         become corrupted.
        ";
}

notification attempt-to-exceed-max-sequence {
    uses notification-instance-hdr;
    leaf lsp-id {
        type lsp-id;
        description
            "LSP ID.";
    }
    description
        "This notification is sent when the system
         wraps the 32-bit sequence counter of an LSP.
        ";
}

notification id-len-mismatch {
    uses notification-instance-hdr;
    uses notification-interface-hdr;

    leaf pdu-field-len {
        type uint8;
        description
            "Size of the ID length in the received PDU";
    }
    leaf raw-pdu {
        type binary;
        description
            "Received raw PDU.";
    }
    description
        "This notification is sent when we receive a PDU
         with a different value for the System ID length.
         The notification generation must be throttled
         with at least a 5 second gap.
        ";
}

notification max-area-addresses-mismatch {
    uses notification-instance-hdr;
    uses notification-interface-hdr;
```



```
    leaf max-area-addresses {
      type uint8;
      description
        "Received number of supported areas";
    }
    leaf raw-pdu {
      type binary;
      description
        "Received raw PDU.";
    }
    description
      "This notification is sent when we receive a PDU
      with a different value for the Maximum Area Addresses.
      The notification generation must be throttled
      with at least a 5 second gap.
      ";
  }

notification own-lsp-purge {
  uses notification-instance-hdr;
  uses notification-interface-hdr;
  leaf lsp-id {
    type lsp-id;
    description
      "LSP ID.";
  }
  description
    "This notification is sent when the system
    receives a PDU with its own system ID and zero age.
    ";
}

notification sequence-number-skipped {
  uses notification-instance-hdr;
  uses notification-interface-hdr;
  leaf lsp-id {
    type lsp-id;
    description
      "LSP ID.";
  }
  description
    "This notification is sent when the system
    receives a PDU with its own system ID and
    different contents. The system has to reissue
    the LSP with a higher sequence number.
    ";
}
```



```
notification authentication-type-failure {
  uses notification-instance-hdr;
  uses notification-interface-hdr;
  leaf raw-pdu {
    type binary;
    description
      "Received raw PDU.";
  }
  description
    "This notification is sent when the system
    receives a PDU with the wrong authentication type
    field.
    The notification generation must be throttled with
    at least a 5 second gap.
    ";
}
```

```
notification authentication-failure {
  uses notification-instance-hdr;
  uses notification-interface-hdr;
  leaf raw-pdu {
    type binary;
    description
      "Received raw PDU.";
  }
  description
    "This notification is sent when the system
    receives a PDU with the wrong authentication
    information.
    The notification generation must be throttled with
    at least a 5 second gap.
    ";
}
```

```
notification version-skew {
  uses notification-instance-hdr;
  uses notification-interface-hdr;
  leaf protocol-version {
    type uint8;
    description
      "Protocol version received in the PDU.";
  }
  leaf raw-pdu {
    type binary;
    description
      "Received raw PDU.";
  }
  description
```



```
    "This notification is sent when the system
    receives a PDU with a different protocol version
    number.
    The notification generation must be throttled with at least
    a 5 second gap.
    ";
}

notification area-mismatch {
    uses notification-instance-hdr;
    uses notification-interface-hdr;
    leaf raw-pdu {
        type binary;
        description
            "Received raw PDU.";
    }
    description
        "This notification is sent when the system
        receives a Hello PDU from an IS that does
        not share any area address.
        The notification generation must be throttled with at least
        a 5 second gap.
        ";
}

notification rejected-adjacency {
    uses notification-instance-hdr;
    uses notification-interface-hdr;
    leaf raw-pdu {
        type binary;
        description
            "Received raw PDU.";
    }
    leaf reason {
        type string;
        description
            "The system may provide a reason to reject the
            adjacency. If the reason is not available,
            the system use an empty string.";
    }
    description
        "This notification is sent when the system
        receives a Hello PDU from an IS but does not
        establish an adjacency for some reason.
        The notification generation must be throttled with at least
        a 5 second gap.
        ";
}
```



```
notification protocols-supported-mismatch {
  uses notification-instance-hdr;
  uses notification-interface-hdr;
  leaf raw-pdu {
    type binary;
    description
      "Received raw PDU.";
  }
  leaf-list protocols {
    type uint8;
    description
      "The list of protocols supported by the
       remote system.";
  }
  description
    "This notification is sent when the system
     receives a non pseudonode LSP that has no matching
     protocol supported.
     The notification generation must be throttled with at least
     a 5 second gap.
    ";
}

notification lsp-error-detected {
  uses notification-instance-hdr;
  uses notification-interface-hdr;
  leaf lsp-id {
    type lsp-id;
    description
      "LSP ID.";
  }
  leaf raw-pdu {
    type binary;
    description
      "Received raw PDU.";
  }
  leaf error-offset {
    type uint32;
    description
      "If the problem is a malformed TLV,
       the error-offset points to the start of the TLV.
       If the problem is with the LSP header,
       the error-offset points to the suspicious byte";
  }
  leaf tlv-type {
    type uint8;
    description
      "if the problem is a malformed TLV, the tlv-type is set
```



```
        to the type value of the suspicious TLV.
        Otherwise this leaf is not present.";
    }
    description
        "This notification is sent when the system
        receives a LSP with a parse error.
        The notification generation must be throttled with at least
        a 5 second gap.
        ";
}

notification adjacency-change {
    uses notification-instance-hdr;
    uses notification-interface-hdr;
    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 system-id;
        description
            "Describes the system-id of the neighbor.";
    }
    leaf level {
        type 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.";
}

notification lsp-received {
  uses notification-instance-hdr;
  uses notification-interface-hdr;

  leaf lsp-id {
    type lsp-id;
    description
      "LSP ID.";
  }
  leaf sequence {
    type uint32;
    description
      "Sequence number of the received LSP.";
  }
  leaf received-timestamp {
    type yang:timestamp;

    description
      "This leaf describes the timestamp
      when the LSP was received. ";
  }
  leaf neighbor-system-id {
    type system-id;
    description
      "Describes the system-id of the neighbor
      that sent the LSP.";
  }
  description
    "This notification is sent when a LSP
    is received.
    The notification generation must be throttled with at least
    a 5 second gap. ";
}
```



```
notification lsp-generation {
  uses notification-instance-hdr;

  leaf lsp-id {
    type lsp-id;
    description
      "LSP ID.";
  }
  leaf sequence {
    type uint32;
    description
      "Sequence number of the received LSP.";
  }
  leaf send-timestamp {
    type yang:timestamp;

    description
      "This leaf describes the timestamp
      when our LSP was regenerated. ";
  }
  description
    "This notification is sent when a LSP
    is regenerated.
    The notification generation must be throttled with at least
    a 5 second gap. ";
}

}

<CODE ENDS>
```

8. IS-IS Segment Routing YANG Module

```
<CODE BEGINS> file "ietf-isis-sr@2016-10-26.yang"

module ietf-isis-sr {
  namespace "urn:ietf:params:xml:ns:"
    + "yang:ietf-isis-sr";
  prefix isis-sr;

  import ietf-routing {
    prefix "rt";
  }

  import ietf-segment-routing-common {
    prefix "sr-cmn";
```



```
}

import ietf-segment-routing {
  prefix "sr";
}

import ietf-isis {
  prefix "isis";
}

organization
  "IETF ISIS Working Group";

contact
  "WG List:  <mailto:spring@ietf.org>;

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

             Acee Lindem
             <mailto:acee@cisco.com>;
             Yingzhen Qu
             <mailto:yiqu@cisco.com>;
             Pushpasis Sarkar
             <mailto:psarkar@juniper.net>;
             Ing-Wher Chen
             <mailto:ing-wher.chen@ericsson.com>;
             Jeff Tantsura
             <mailto:jeff.tantsura@ericsson.com>;

  ";

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

revision 2016-10-26 {
  description
    "Initial revision.";
  reference "draft-ietf-isis-yang-isis-cfg-13";
}

/* Identities */
```



```
/* Features */

feature remote-lfa-sr {
    description
        "Enhance rLFA to use SR path.";
}

feature ti-lfa {
    description
        "Enhance IPFRR with ti-lfa
        support";
}

/* Groupings */

grouping adjacency-state {
    description
        "This group will extend adjacency state.";
    list adjacency-sid {
        key value;
        leaf af {
            type identityref {
                base rt:address-family;
            }
            description
                "Address-family associated with the
                segment ID";
        }
        leaf value {
            type uint32;
            description
                "Value of the Adj-SID.";
        }
        leaf weight {
            type uint8;
            description
                "Weight associated with
                the adjacency SID.";
        }
        leaf protection-requested {
            type boolean;
            description
                "Describe if the adjacency SID
                must be protected.";
        }
    }
    description
        "List of adjacency Segment IDs.";
}
```



```
}
```

```
grouping prefix-segment-id {  
  description  
    "This group defines segment routing extensions  
    for prefixes.";  
  
  list sid-list {  
    key value;  
  
    leaf flags {  
      type bits {  
        bit readvertisement {  
          position 7;  
          description  
            "If set, then the prefix to  
            which this Prefix-SID is attached,  
            has been propagated by the  
            router either from another level  
            or from redistribution.";  
        }  
  
        bit php {  
          position 5;  
          description  
            "If set, then the penultimate hop MUST NOT  
            pop the Prefix-SID before delivering the packet  
            to the node  
            that advertised the Prefix-SID.";  
        }  
      }  
  
      bit explicit-null {  
        position 4;  
        description  
          "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.";  
      }  
  
      bit value {  
        position 3;  
        description  
          "If set, then the Prefix-SID carries a  
          value (instead of an index).  
          By default the flag is UNSET.";  
      }  
    }  
  }  
}
```



```
    }
    bit local {
        position 2;
        description
            "If set, then the value/index carried by
            the Prefix-SID has local significance.
            By default the flag is UNSET.";
    }
}
description
    "Describes flags associated with the
    segment ID.";
}

leaf algorithm {
    type uint8;
    description
        "Algorithm to be used for path computation.";
}
leaf value {
    type uint32;
    description
        "Value of the prefix-SID.";
}
description
    "List of segments.";
}
}

grouping adjacency-segment-id {
    description
        "This group defines segment routing extensions
        for adjacencies.";

    list sid-list {
        key value;

        leaf flags {
            type bits {
                bit address-family {
                    position 7;
                    description
                        "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.";
                }
            }
            bit backup {
```



```
        position 6;
        description
            "If set, the Adj-SID refers to an
            adjacency being protected
            (e.g.: using IPFRR or MPLS-FRR)";
    }
    bit value {
        position 5;
        description
            "If set, then the SID carries a
            value (instead of an index).
            By default the flag is SET.";
    }

    bit local {
        position 4;
        description
            "If set, then the value/index carried by
            the SID has local significance.
            By default the flag is SET.";
    }
    bit set {
        position 3;
        description
            "When set, the S-Flag indicates that the
            Adj-SID refers to a set of adjacencies";
    }
}

description
    "Describes flags associated with the
    segment ID.";
}
leaf weight {
    type uint8;
    description
        "The value represents the weight of the Adj-SID
        for the purpose of load balancing.";
}
leaf neighbor-id {
    type isis:system-id;
    description
        "Describes the system ID of the neighbor
        associated with the SID value. This is only
        used on LAN adjacencies.";
}
leaf value {
```



```
        type uint32;
        description
            "Value of the Adj-SID.";
    }
    description
        "List of segments.";
}

}
grouping segment-routing-binding-tlv {
list segment-routing-bindings {

    key "fec range";

    leaf fec {
        type string;
        description
            "IP (v4 or v6) range to be bound to SIDs.";
    }

    leaf range {
        type uint16;
        description
            "Describes number of elements to assign
            a binding to.";
    }

    leaf flags {
        type bits {
            bit address-family {
                position 7;
                description
                    "If unset, then the Prefix FEC
                    carries an IPv4 Prefix.
                    If set then the Prefix FEC carries an
                    IPv6 Prefix.";
            }
            bit mirror {
                position 6;
                description
                    "Set if the advertised SID/path
                    corresponds to a mirrored context.
                    ";
            }
            bit flooding {
                position 5;
                description
                    "If the S bit is set(1),
```



```
    the IS-IS Router CAPABILITY TLV
    MUST be flooded across the entire routing domain.
    If the S bit is
    not set(0), the TLV MUST NOT be leaked between levels.
    This bit MUST NOT be altered during the TLV leaking.";
  }
  bit down {
    position 4;
    description
    "When the IS-IS Router CAPABILITY TLV is
    leaked from level-2 to level-1, the D bit
    MUST be set. Otherwise, this bit MUST
    be clear. IS-IS Router capability TLVs
    with the D bit set MUST NOT
    be leaked from level-1 to level-2.
    This is to prevent TLV looping.
    ";
  }
  bit attached {
    position 3;
    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.
    ";
  }
}
description
  "Flags of the binding.";
}

leaf weight {
  type uint8;
  description
    "Weight of the path for loadbalancing purpose.";
}

container binding {
  container prefix-sid {
    uses prefix-segment-id;
    description
      "Binding prefix SID to the range.";
  }
  leaf ero-metric {
```



```
    type uint32;
    description
      "Cost of ERO path.";
  }
  container ero {
    leaf address-family {
      type identityref {
        base rt:address-family;
      }
      description
        "Address-family.";
    }

    leaf loose {
      type boolean;
      description
        "Set to true,
        if hop is a loose hop.";
    }

    leaf address {
      type string;
      description
        "IP address of a node on the
        path.";
    }

    description
      "Binding ERO path to the range.";
  }
  container backup-ero {
    leaf address-family {
      type identityref {
        base rt:address-family;
      }
      description
        "Address-family.";
    }

    leaf loose {
      type boolean;
      description
        "Set to true,
        if hop is a loose hop.";
    }

    leaf address {
      type string;
      description
        "IP address of a node on the
```



```
        path.";
    }

    description
        "Binding backup ERO path to the range.";
}
container unnumbered-interface-id-ero {
    leaf router-id {
        type string;
        description
            "Router ID of the node owning the interface.";
    }
    leaf interface-id {
        type uint32;
        description
            "Interface ID on which the path is built.";
    }
    description
        "Binding a path over unnumbered interface.";
}
container backup-unnumbered-interface-id-ero {
    leaf router-id {
        type string;
        description
            "Router ID of the node owning the interface.";
    }
    leaf interface-id {
        type uint32;
        description
            "Interface ID on which the path is built.";
    }
    description
        "Binding a backup path over unnumbered interface.";
}
description
    "Bindings associated with the range.";
}

description
    "This container describes list of SID/Label
    bindings.
    ISIS reference is TLV 149.";
}
description
    "Defines binding TLV for database.";
}
/* Cfg */
```



```
augment "/rt:routing/" +
  "rt:control-plane-protocols/rt:control-plane-protocol"+
  "/isis:isis" {
  when "/rt:routing/rt:control-plane-protocols/"+
    "rt:control-plane-protocol/rt:type = 'isis:isis'" {
    description
      "This augment ISIS routing protocol when used";
  }
  description
    "This augments ISIS protocol configuration
    with segment routing.";

  uses sr:controlplane-cfg;
  container protocol-srgb {
    if-feature sr:protocol-srgb;
    uses sr-cmn:srgb-cfg;
    description
      "Per-protocol SRGB.";
  }
}
```

```
augment "/rt:routing/" +
  "rt:control-plane-protocols/rt:control-plane-protocol"+
  "/isis:isis/isis:interfaces/isis:interface" {
  when "/rt:routing/rt:control-plane-protocols/"+
    "rt:control-plane-protocol/rt:type = 'isis:isis'" {
    description
      "This augment ISIS routing protocol when used";
  }
  description
    "This augments ISIS protocol configuration
    with segment routing.";

  uses sr:igp-interface-cfg;
}
```

```
augment "/rt:routing/" +
  "rt:control-plane-protocols/rt:control-plane-protocol"+
  "/isis:isis/isis:interfaces/isis:interface"+
  "/isis:fast-reroute" {
  when "/rt:routing/rt:control-plane-protocols/"+
    "rt:control-plane-protocol/rt:type = 'isis:isis'" {
    description
      "This augment ISIS routing protocol when used";
  }
  description
```



```
    "This augments ISIS IP FRR with TILFA.";

    container ti-lfa {
        if-feature ti-lfa;
        leaf enable {
            type boolean;
            description
                "Enables TI-LFA computation.";
        }
        description
            "TILFA configuration.";
    }
}

augment "/rt:routing/" +
    "rt:control-plane-protocols/rt:control-plane-protocol"+
    "/isis:isis/isis:interfaces/isis:interface"+
    "/isis:fast-reroute/isis:lfa/isis:remote-lfa" {
    when "/rt:routing/rt:control-plane-protocols/" +
        "rt:control-plane-protocol/rt:type = 'isis:isis'" {
        description
            "This augment ISIS routing protocol when used";
    }
    description
        "This augments ISIS remoteLFA config with
        use of segment-routing path.";

    leaf use-segment-routing-path {
        if-feature remote-lfa-sr;
        type boolean;
        description
            "force remote LFA to use segment routing
            path instead of LDP path.";
    }
}

}

/* Operational states */

augment "/rt:routing-state/" +
    "rt:control-plane-protocols/rt:control-plane-protocol"+
    "/isis:isis" {
    when "/rt:routing-state/rt:control-plane-protocols/" +
        "rt:control-plane-protocol/rt:type = 'isis:isis'" {
        description
```



```
    "This augment ISIS routing protocol when used";
  }
  description
    "This augments ISIS protocol configuration
    with segment routing.";

  uses sr:controlplane-cfg;
  container protocol-srgb {
    if-feature sr:protocol-srgb;
    uses sr-cmn:srgb-cfg;
    description
      "Per-protocol SRGB.";
  }
}

augment "/rt:routing-state/" +
  "rt:control-plane-protocols/rt:control-plane-protocol"+
  "/isis:isis/isis:interfaces/isis:interface" {
  when "/rt:routing-state/rt:control-plane-protocols/" +
    "rt:control-plane-protocol/rt:type = 'isis:isis'" {
    description
      "This augment ISIS routing protocol when used";
  }
  description
    "This augments ISIS protocol configuration
    with segment routing.";

  uses sr:igp-interface-cfg;
}

augment "/rt:routing-state/" +
  "rt:control-plane-protocols/rt:control-plane-protocol"+
  "/isis:isis/isis:interfaces/isis:interface" +
  "/isis:adjacencies/isis:adjacency" {
  when "/rt:routing-state/rt:control-plane-protocols/" +
    "rt:control-plane-protocol/rt:type = 'isis:isis'" {
    description
      "This augment ISIS routing protocol when used";
  }
  description
    "This augments ISIS protocol configuration
    with segment routing.";

  uses adjacency-state;
}
```



```
augment "/rt:routing-state/" +
  "rt:control-plane-protocols/rt:control-plane-protocol"+
  "/isis:isis/isis:database/isis:level-db/isis:lsp"+
  "/isis:extended-is-neighbor/isis:neighbor" {
  when "/rt:routing-state/rt:control-plane-protocols/"+
    "rt:control-plane-protocol/rt:type = 'isis:isis'" {
    description
      "This augment ISIS routing protocol when used";
  }
  description
    "This augments ISIS protocol LSDB neighbor.";
  uses adjacency-segment-id;
}
augment "/rt:routing-state/" +
  "rt:control-plane-protocols/rt:control-plane-protocol"+
  "/isis:isis/isis:database/isis:level-db/isis:lsp"+
  "/isis:mt-is-neighbor/isis:neighbor" {
  when "/rt:routing-state/rt:control-plane-protocols/"+
    "rt:control-plane-protocol/rt:type = 'isis:isis'" {
    description
      "This augment ISIS routing protocol when used";
  }
  description
    "This augments ISIS protocol LSDB neighbor.";
  uses adjacency-segment-id;
}
augment "/rt:routing-state/" +
  "rt:control-plane-protocols/rt:control-plane-protocol"+
  "/isis:isis/isis:database/isis:level-db/isis:lsp"+
  "/isis:extended-ipv4-reachability/isis:prefixes" {
  when "/rt:routing-state/rt:control-plane-protocols/"+
    "rt:control-plane-protocol/rt:type = 'isis:isis'" {
    description
      "This augment ISIS routing protocol when used";
  }
  description
    "This augments ISIS protocol LSDB prefix.";
  uses prefix-segment-id;
}
augment "/rt:routing-state/" +
  "rt:control-plane-protocols/rt:control-plane-protocol"+
  "/isis:isis/isis:database/isis:level-db/isis:lsp"+
  "/isis:mt-extended-ipv4-reachability/isis:prefixes" {
  when "/rt:routing-state/rt:control-plane-protocols/"+
    "rt:control-plane-protocol/rt:type = 'isis:isis'" {
```



```
    description
    "This augment ISIS routing protocol when used";
  }
  description
  "This augments ISIS protocol LSDB prefix.";
  uses prefix-segment-id;
}
augment "/rt:routing-state/" +
  "rt:control-plane-protocols/rt:control-plane-protocol"+
  "/isis:isis/isis:database/isis:level-db/isis:lsp"+
  "/isis:ipv6-reachability/isis:prefixes" {
    when "/rt:routing-state/rt:control-plane-protocols/" +
    "rt:control-plane-protocol/rt:type = 'isis:isis'" {
    description
    "This augment ISIS routing protocol when used";
  }
  description
  "This augments ISIS protocol LSDB prefix.";
  uses prefix-segment-id;
}
augment "/rt:routing-state/" +
  "rt:control-plane-protocols/rt:control-plane-protocol"+
  "/isis:isis/isis:database/isis:level-db/isis:lsp"+
  "/isis:mt-ipv6-reachability/isis:prefixes" {
    when "/rt:routing-state/rt:control-plane-protocols/" +
    "rt:control-plane-protocol/rt:type = 'isis:isis'" {
    description
    "This augment ISIS routing protocol when used";
  }
  description
  "This augments ISIS protocol LSDB prefix.";
  uses prefix-segment-id;
}
augment "/rt:routing-state/" +
  "rt:control-plane-protocols/rt:control-plane-protocol"+
  "/isis:isis/isis:database/isis:level-db/isis:lsp" {
  when "/rt:routing-state/rt:control-plane-protocols/" +
  "rt:control-plane-protocol/rt:type = 'isis:isis'" {
  description
  "This augment ISIS routing protocol when used";
  }
  description
  "This augments ISIS protocol LSDB.";
  uses segment-routing-binding-tlv;
```



```
}

/* Notifications */

}

<CODE ENDS>
```

9. Security Considerations

Configuration and state data defined in this document are designed to be accessed via the NETCONF protocol [[RFC6241](#)].

As IS-IS is an IGP protocol (critical piece of the network), ensuring stability and security of the protocol is mandatory for the network service.

Authors recommends to implement NETCONF access control model ([[RFC6536](#)]) to restrict access to all or part of the configuration to specific users. Access control to RPCs is also critical as RPC allows to clear protocol datastructures that would definitively impact the network service. This kind of RPC needs only to be used in specific cases by well-known experienced users.

Authors consider that all the configuration is considered as sensitive/vulnerable as well as RPCs. But security teams can decide to open some part of the configuration to less experienced users depending on the internal organization, for example:

- o User FullWrite: would access to the whole data model. This kind of profile may be restricted to few experienced people.
- o User PartialWrite: would only access to configuration part within /isis/interfaces/interface. So this kind of profile is restricted to creation/modification/deletion of interfaces. This profile does not have access to RPC.
- o User Read: would only access to state part /isis-state.

Unauthorized access to configuration or RPC may cause high damages to the network service.

The /isis-state/database may contain authentication information. As presented in the description of the /isis-state/database/level-1/lsp/authentication/authentication-key, the authentication MUST never be presented in plaintext format for security reason. Authors

recommend the usage of MD5 to display or return the authentication-key.

Some authentication-key may also be present in the /isis configuration. When configuring IS-IS using the NETCONF protocol, authors recommends the usage of secure transport of NETCONF using SSH ([RFC6242]).

10. Contributors

Authors would like to thank Kiran Agrahara Sreenivasa, Dean Bogdanovic, Yingzhen Qu, Yi Yang for their major contributions to the draft.

11. Acknowledgements

TBD.

12. IANA Considerations

The IANA is requested to assign two new URIs from the IETF XML registry ([RFC3688]). Authors are suggesting the following URIs :

URI: urn:ietf:params:xml:ns:yang:ietf-isis
Registrant Contact: IS-IS WG
XML: N/A, the requested URI is an XML namespace

URI: urn:ietf:params:xml:ns:yang:ietf-isis-sr
Registrant Contact: IS-IS WG
XML: N/A, the requested URI is an XML namespace

This document also requests two new YANG modules name in the YANG Module Names registry ([RFC6020]) with the following suggestion :

name: ietf-isis
namespace: urn:ietf:params:xml:ns:yang:ietf-isis
prefix: isis
reference: RFC XXXX

name: ietf-isis-sr
namespace: urn:ietf:params:xml:ns:yang:ietf-isis-sr
prefix: isis-sr
reference: RFC XXXX

13. Change log for ietf-isis-sr YANG module**13.1. From version -12 to version -13**

- o Align with new segment routing common module.

13.2. From version -09 to version -11

- o Fixed XPATH in 'when' expressions.

13.3. From version -08 to version -09

- o Align to [draft-ietf-netmod-routing-cfg-23](#).

13.4. From version -07 to version -08

- o Align to [draft-ietf-netmod-routing-cfg-21](#).

14. Change log for ietf-isis YANG module**14.1. From version -12 to version -13**

- o Move feature nlpid-control to container rather than list.
- o Rename multi-topology to topologies to align with OSPF.
- o Rename bfd/enabled to bfd/enable for consistency reason.
- o Add support for NSR with a feature.

14.2. From version -09 to version -12

- o Rename node-tag container to node-tags.

14.3. From version -08 to version -09

- o Added container before af list.
- o Added container before topology list.
- o Aligned LFA if per level cfg.
- o Align to [draft-ietf-netmod-routing-cfg-23](#).

14.4. From version -07 to version -08

- o Remove selector from system-id type.
- o Add some default values.
- o Moved lists to containers+groupings for per level configuration.
- o remove routing-instance as per core routing model v21.
- o added BFD leaf (no more BFD protocol model).
- o changed keychain module reference.

14.5. From version -05 to version -07

- o Move Overload config from list to container.
- o Move Overload-max-metric config from list to container.
- o Move preference config from list to container.
- o Add Node flag in config.
- o Removed BFD config => moved to isis-bfd module.
- o Remove call to routing policy model.

14.6. From version -03 to version -05

- o Correct invalid references to previous versions of core routing model.
- o Remove BFD config and replace by groupings from ietf-bfd.
- o Adding routing-policy support through routing-policy model.

14.7. From version -02 to version -03

- o Reviewed config and op state groupings.
- o Add default value to lfa candidate-disabled.
- o Add enable leaf to isis container to reflect admin state.
- o Move to VRF centric only.
- o Segment routing is part of a separate module.

14.8. From version -01 to version -02

- o Adding IPFRR.
- o Adding igp-ldp-sync.
- o Adding segment-routing.
- o Adding instance reference to operational states.
- o Move AF type from string to identity.
- o Updated router-capability in LSDB description.
- o packet counters moved to interface-packet-counters.
- o Added modification information in lsp-log.
- o Removing igp-ldp-sync timer in IS-IS.
- o Defining hierarchy for operational states.
- o Adding clns-mtu.
- o Adding key-chain.

14.9. From version -00 to version -01

- o Interface metric move from af container to interface container.
- o Hello-padding on interface moved to hello-padding-disable with empty type.
- o three-way-handshake removed.
- o route preference changed to a choice.
- o csnp-authentication/psnp-authentication merged to authentication container.
- o lsp-gen-interval-exp-delay removed.
- o Added overload-max-metric feature.
- o overload-max-metric is in a separate container.
- o Change hello-padding to container.

- o Change bfd to container.
- o Make BFD a feature.
- o Create mpls-te container and put router-id inside.
- o Remove GR helper disable and timers.

15. Normative References

- [I-D.ietf-netmod-routing-cfg]
Lhotka, L. and A. Lindem, "A YANG Data Model for Routing Management", [draft-ietf-netmod-routing-cfg-24](#) (work in progress), October 2016.
- [I-D.ietf-rtgwg-yang-key-chain]
Lindem, A., Qu, Y., Yeung, D., Chen, I., Zhang, Z., and Y. Yang, "Routing Key Chain YANG Data Model", [draft-ietf-rtgwg-yang-key-chain-09](#) (work in progress), September 2016.
- [I-D.ietf-spring-sr-yang]
Litkowski, S., Qu, Y., Sarkar, P., and J. Tantsura, "YANG Data Model for Segment Routing", [draft-ietf-spring-sr-yang-04](#) (work in progress), October 2016.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/[RFC2119](#), March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", [BCP 81](#), [RFC 3688](#), DOI 10.17487/RFC3688, January 2004, <<http://www.rfc-editor.org/info/rfc3688>>.
- [RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)", [RFC 5120](#), DOI 10.17487/[RFC5120](#), February 2008, <<http://www.rfc-editor.org/info/rfc5120>>.
- [RFC5286] Atlas, A., Ed. and A. Zinin, Ed., "Basic Specification for IP Fast Reroute: Loop-Free Alternates", [RFC 5286](#), DOI 10.17487/RFC5286, September 2008, <<http://www.rfc-editor.org/info/rfc5286>>.

- [RFC5443] Jork, M., Atlas, A., and L. Fang, "LDP IGP Synchronization", [RFC 5443](#), DOI 10.17487/RFC5443, March 2009, <<http://www.rfc-editor.org/info/rfc5443>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", [RFC 6020](#), DOI 10.17487/RFC6020, October 2010, <<http://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", [RFC 6241](#), DOI 10.17487/RFC6241, June 2011, <<http://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", [RFC 6242](#), DOI 10.17487/RFC6242, June 2011, <<http://www.rfc-editor.org/info/rfc6242>>.
- [RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", [RFC 6536](#), DOI 10.17487/RFC6536, March 2012, <<http://www.rfc-editor.org/info/rfc6536>>.
- [RFC7490] Bryant, S., Filsfils, C., Previdi, S., Shand, M., and N. So, "Remote Loop-Free Alternate (LFA) Fast Reroute (FRR)", [RFC 7490](#), DOI 10.17487/RFC7490, April 2015, <<http://www.rfc-editor.org/info/rfc7490>>.

Appendix A. Example of IS-IS configuration in XML

This section gives an example of configuration of an IS-IS instance on a device. The example is written in XML.

```
<?xml version="1.0" encoding="utf-8"?>
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <routing xmlns="urn:ietf:params:xml:ns:yang:ietf-routing">
    <name>SLI</name>
    <router-id>1.1.1.1</router-id>
    <description/>
    <interfaces>
      <interface>
        <name>Loopback0</name>
      </interface>
      <interface>
        <name>Eth1</name>
      </interface>
    </interfaces>
    <control-plane-protocols>
```



```
<control-plane-protocol>
  <name>ISIS</name>
  <description/>
  <type>isis:isis</type>
  <isis xmlns="urn:ietf:params:xml:ns:yang:ietf-isis">
    <enable>true</enable>
    <level-type>level-2</level-type>
    <system-id>87FC.FCDF.4432</system-id>
    <area-address>49.0001</area-address>
    <mpls-te>
      <ipv4-router-id>1.1.1.1</ipv4-router-id>
    </mpls-te>
    <lsp-lifetime>65535</lsp-lifetime>
    <lsp-refresh>65000</lsp-refresh>
    <metric-type>
      <value>wide</value>
    </metric-type>
    <default-metric>
      <value>111111</value>
    </default-metric>
    <afs>
      <af>
        <af>ipv4-unicast</af>
        <enabled>true</enabled>
      </af>
    </afs>
    <interfaces>
      <interface>
        <name>Loopback0</name>
        <tag>200</tag>
        <metric>
          <value>0</value>
        </metric>
        <passive>true</passive>
      </interface>
      <interface>
        <name>Eth1</name>
        <level-type>level-2</level-type>
        <interface-type>point-to-point</interface-type>
        <metric>
          <value>167890</value>
        </metric>
      </interface>
    </interfaces>
  </isis>
</control-plane-protocol>
</control-plane-protocols>
</routing>
```



```
<interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
  <interface>
    <name>Loopback0</name>
    <description/>
    <type/>
    <link-up-down-trap-enable/>
    <ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
      <mtu/>
      <address>
        <ip>1.1.1.1</ip>
        <prefix-length>32</prefix-length>
      </address>
    </ipv4>
  </interface>
  <interface>
    <name>Eth1</name>
    <description/>
    <type/>
    <link-up-down-trap-enable/>
    <ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
      <mtu/>
      <address>
        <ip>10.0.0.1</ip>
        <prefix-length>30</prefix-length>
      </address>
    </ipv4>
  </interface>
</interfaces>
</data>
```

Authors' Addresses

Stephane Litkowski
Orange

Email: stephane.litkowski@orange.com

Derek Yeung
Arrcus, Inc

Email: derek@arrcus.com

Acee Lindem
Cisco Systems

Email: acee@cisco.com

Jeffrey Zhang
Juniper Networks

Email: zzhang@juniper.net

Ladislav Lhotka
CZ.NIC

Email: lhotka@nic.cz