

SPRING Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: September 6, 2015

S. Litkowski  
Orange Business Service  
A. Lindem  
Cisco Systems  
P. Sarkar  
Juniper Networks  
I. Chen  
Ericsson  
March 05, 2015

YANG Data Model for Segment Routing  
draft-litkowski-spring-sr-yang-00

## Abstract

This document defines a YANG data model for segment routing configuration and operation. This YANG model is intended to be used on network elements to configure or operate segment routing.

## 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 September 6, 2015.

## Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

Internet-Draft

sr-yang-cfg

March 2015

This document is subject to [BCP 78](http://trustee.ietf.org/license-info) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">2</a>
<a href="#">1.1.</a>	Tree diagram . . . . .	<a href="#">2</a>
<a href="#">2.</a>	Design of the Data Model . . . . .	<a href="#">3</a>
<a href="#">3.</a>	Configuration . . . . .	<a href="#">5</a>
<a href="#">3.1.</a>	Adjacency SID properties . . . . .	<a href="#">5</a>
<a href="#">3.1.1.</a>	Bundling . . . . .	<a href="#">5</a>
<a href="#">3.1.2.</a>	Protection . . . . .	<a href="#">6</a>
<a href="#">3.2.</a>	Prefix SID properties . . . . .	<a href="#">6</a>
<a href="#">4.</a>	Control plane configuration . . . . .	<a href="#">7</a>
<a href="#">5.</a>	States . . . . .	<a href="#">7</a>
<a href="#">6.</a>	Notifications . . . . .	<a href="#">7</a>
<a href="#">7.</a>	YANG Module . . . . .	<a href="#">8</a>
<a href="#">8.</a>	Security Considerations . . . . .	<a href="#">19</a>
<a href="#">9.</a>	Acknowledgements . . . . .	<a href="#">19</a>
<a href="#">10.</a>	IANA Considerations . . . . .	<a href="#">20</a>
<a href="#">11.</a>	Normative References . . . . .	<a href="#">20</a>
	Authors' Addresses . . . . .	<a href="#">21</a>

## [1.](#) Introduction

This document defines a YANG data model for segment routing configuration and operation.

### [1.1.](#) Tree diagram

A simplified graphical representation of the data model is presented in [Section 2](#).

The meaning of the symbols in these diagrams is as follows:

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

- o Curly braces "{" and "}" contain names of optional features that make the corresponding node conditional.

- o Abbreviations before data node names: "rw" means configuration (read-write), and "ro" state data (read-only).
- o Symbols after data node names: "?" means an optional node and "\*" denotes a "list" or "leaf-list".
- o Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").
- o Ellipsis ("...") stands for contents of subtrees that are not shown.

## 2. Design of the Data Model

This is the initial version of this module and its relationship to the protocol modules. It is expected that there will be changes as the module matures.

```

module: ietf-segment-routing
augment /rt:routing/rt:routing-instance:
  +--rw segment-routing
    +--rw transport-type?  identityref
    +--rw bindings
      |  +--rw mapping-server {mapping-server}?
      |  |  +--rw ipv4
      |  |  |  +--rw mapping-entry* [prefix]
      |  |  |  |  +--rw prefix      inet:ipv4-prefix
      |  |  |  |  +--rw start-sid?  uint32
      |  |  |  |  +--rw range?     uint32
      |  |  +--rw ipv6
      |  |  |  +--rw mapping-entry* [prefix]
      |  |  |  |  +--rw prefix      inet:ipv6-prefix
      |  |  |  |  +--rw start-sid?  uint32
      |  |  |  |  +--rw range?     uint32
      |  +--rw srgb* [lower-bound upper-bound]
      |  |  +--rw lower-bound    uint32

```

```

|   +--rw upper-bound      uint32
+--rw interfaces
  +--rw interface* [name]
    +--rw name              if:interface-ref
    +--rw adjacency-sid
      |   +--rw advertise-adj-group-sid* [group-id]
      |   |   +--rw group-id      uint32
      |   |   +--rw advertise-protection?      enumeration
    +--rw prefix-sid
      +--rw ipv4
        |   +--rw prefix-sid* [value]
        |   |   +--rw value-type?      enumeration

```

```

|   +--rw value              uint32
|   +--rw node-flag?         boolean
|   +--rw last-hop-behavior? enumeration
+--rw ipv6
  +--rw prefix-sid* [value]
    +--rw value-type?      enumeration
    +--rw value            uint32
    +--rw node-flag?       boolean
    +--rw last-hop-behavior? enumeration
augment /rt:routing/rt:routing-instance/rt:routing-protocols/rt:routing-protoco
  +--rw segment-routing
    +--rw enabled?         boolean
    +--rw bindings
      +--rw advertise?     boolean
      +--rw receive?       boolean
augment /rt:routing/rt:routing-instance/rt:routing-protocols/rt:routing-protoco
  +--rw segment-routing
    +--rw enabled?         boolean
    +--rw bindings
      +--rw advertise?     boolean
      +--rw receive?       boolean
augment /rt:routing-state/rt:routing-instance:
  +--ro segment-routing
    +--ro label-blocks*
      |   +--ro lower-bound?  uint32
      |   +--ro upper-bound?  uint32
      |   +--ro size?         uint32
      |   +--ro free?         uint32
      |   +--ro used?         uint32

```

```

    +---ro global-sid-list
      +---ro sid* [target sid source source-protocol binding-type]
        +---ro target          string
        +---ro sid              uint32
        +---ro algorithm?      uint8
        +---ro source           inet:ip-address
        +---ro used?            boolean
        +---ro source-protocol  leafref
        +---ro binding-type     enumeration
notifications:
  +---n segment-routing-global-sid-collision
  | +---ro received-target?    string
  | +---ro original-target?    string
  | +---ro index?              uint32
  | +---ro routing-protocol?   leafref
  +---n segment-routing-index-out-of-range
  +---ro received-target?      string
  +---ro received-index?       uint32
  +---ro routing-protocol?     leafref

```

### [3.](#) Configuration

This module augments the `"/rt:routing/rt:routing-instance:"` with a segment-routing container. This container defines all the configuration parameters related to segment-routing for this particular routing-instance.

The segment-routing configuration is split in global routing-instance configuration and interface configuration.

The global configuration includes :

- o segment-routing transport type : The underlying transport type for segment routing. The version of the model limits the transport type to an MPLS dataplane. The transport-type is only defined once for a particular routing-instance and is agnostic to the control plane used. Only a single transport-type is supported in this version of the model.
- o bindings : Defines how external information is mapped to a segment ID. The current version supports a mapping-server where static prefix-to-SID bindings can be defined. Configuration of bindings

does not allow advertisement of those bindings. Advertisement must be controlled by each routing-protocol instance.

- o SRGB (Segment Routing Global Block): Defines a list of label blocks represented by a pair of lower-bound/upper-bound labels. The SRGB is also agnostic to the control plane used. So all routing-protocol instance will have to advertise the same SRGB.

The interface configuration includes :

- o Adjacency SID properties
- o Prefix SID properties

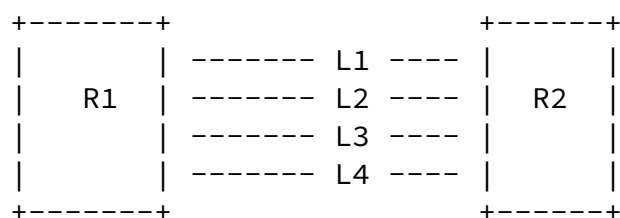
### [3.1.](#) Adjacency SID properties

#### [3.1.1.](#) Bundling

This section is a first proposal on how to use S-bit in Adj-SID to create bundles. Authors would like to trigger discussion based on this first proposal.

In case of parallel IP links between routers, an additional Adjacency SID may be advertised representing more than one adjacency (i.e., a bundle of adjacencies). The "advertise-adj-group-sid" configuration controls whether or not an additional adjacency SID is advertised.

The "advertise-adj-group-sid" would be a list of "group-id". The "group-id" will permit to identify interfaces that must be bundled together.



In the figure above, R1 and R2 are interconnected by four links. A routing protocol adjacency is established on each link. Operator would like to create segment-routing Adj-SID that represent some

bundles of links. We can imagine two different bundles : L1/L2 and L2/L3. To achieve this behavior, the service provider will configure a "group-id" X for both interfaces L1 and L2 and a "group-id" Y for both interfaces L3 and L3. This will result in R1 advertising an additional Adj-SID for each adjacency, for example a Adj-SID with S flag set and value of 400 will be added to L1 and L2. A Adj-SID with S flag set and value of 500 will be added to L3 and L4. As L1/L2 and L3/L4 does not share the same "group-id", a different SID value will be allocated.

### [3.1.2.](#) Protection

The "advertise-protection" defines how protection for an interface is advertised. It does not control the activation or deactivation of protection. If the "single" option is used, a single Adj-SID will be advertised for the interface. If the interface is protected, the B-Flag for the Adj-SID advertisement will be set. If the "dual" option is used and if the interface is protected, two Adj-SIDs will be advertised for the interface adjacencies. One Adj-SID will always have the B-Flag set and the other will have the B-Flag clear. This option is intended to be used in the case of traffic engineering where a path must use either protected segments or non-protected segments.

### [3.2.](#) Prefix SID properties

An interface may have associated IP prefixes. By default, no Prefix-SID will be advertised for any IP prefix associated with an interface.

The operator can control the advertisement of IP prefixes by setting "prefix-sid" in the interface configuration.

The operator can control advertisement of Prefix-SID independently for IPv4 and IPv6. When specified, the "prefix-sid" value must be included.

The value can be expressed as an index (default), or an absolute value. The operator can also control if the "node-flag" is set for the prefix. As the network device owns the prefix, the default is to advertise the prefix with the "node-flag" set.

The "last-hop-behavior" configuration dictates the PHP behavior: "explicit-null", "php", or "non-php".

#### [4.](#) Control plane configuration

Activation of segment-routing extensions for a particular control plane is done by augmenting routing-protocol configuration with segment-routing.

The "enabled" leaf enables segment-routing extensions for the routing-protocol instance.

The "bindings" container controls the routing-protocol instance's advertisement of local bindings and the processing of received bindings.

This model supports ISIS ([\[I-D.ietf-isis-segment-routing-extensions\]](#)) and OSPF as controlplane ([\[I-D.ietf-ospf-segment-routing-extensions\]](#) and [\[I-D.psenak-ospf-segment-routing-ospfv3-extension\]](#)) for segment-routing.

#### [5.](#) States

The operational states contains information reflecting the usage of allocated SRGB labels.

It also includes a list of all global SIDs, their associated bindings, and other information such as the source protocol and algorithm.

#### [6.](#) Notifications

The model proposes two notifications for segment-routing.

- o segment-routing-global-sid-collision: Raised when a control plane advertised index is already associated with another target (in this version, the only defined targets are IPv4 and IPv6 prefixes).

- o segment-routing-index-out-of-range: Raised when a control plane



advertised index fall outside the range of SRGBs configured for the network device.

## 7. YANG Module

```
<CODE BEGINS> file "ietf-segment-routing@2015-03-04.yang"
```

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

  import ietf-inet-types {
    prefix "inet";
  }

  import ietf-routing {
    prefix "rt";
  }

  import ietf-interfaces {
    prefix "if";
  }

  import ietf-isis {
    prefix "isis";
  }

  import ospf {
    prefix "ospf";
  }

  organization
    "IETF SPRING Working Group";

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

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

               Acee Lindem
               <mailto:acee@cisco.com>
               Pushpasis Sarkar
               <mailto:psarkar@juniper.net>
               Ing-Wher Chen
               <mailto:ing-wher.chen@ericsson.com>
```

---

```
    ";

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

revision 2015-02-27 {
  description "Initial";
  reference "draft-litkowski-spring-sr-yang-00";
}

/* Identities */
identity segment-routing-transport {
  description
    "Base identity for segment routing transport.";
}
identity segment-routing-transport-mpls {
  base segment-routing-transport;
  description
    "This identity represents MPLS transport for segment
    routing.";
}

/* Features */

feature mapping-server {
  description
    "Support of SRMS.";
}

/* Groupings */

grouping controlplane-cfg {
  container segment-routing {
    leaf enabled {
      type boolean;
      default false;
      description
        "Enables segment-routing
        protocol extensions.";
    }
  }
  container bindings {
    leaf advertise {
      type boolean;
      default true;
    }
  }
}
```

description  
"Authorize the advertise

Internet-Draft

sr-yang-cfg

March 2015

```
        of local mappings in binding TLV.";
    }
    leaf receive {
        type boolean;
        default true;
        description
            "Authorize the reception and usage
            of binding TLV.";
    }
    description
        "Control of binding advertisement
        and reception.";
}

description
    "segment routing global config.";
}
description
    "Defines protocol configuration.";
}

grouping prefix-sid-cfg {
    list prefix-sid {
        key value;
        leaf value-type {
            type enumeration {
                enum index {
                    description
                        "The value will be
                        interpreted as an index.";
                }

                enum absolute {

                    description
                        "The value will become
                        interpreted as an absolute
                        value.";
                }
            }
        }
    }
}
```

```

    }
    default index;
    description
        "This leaf defines how value
        must be interpreted.";
}

leaf value {
    type uint32;

```

```

    mandatory true;
    description
        "Value associated with
        prefix. The value must
        be interpreted in the
        context of value-type.";
}
leaf node-flag {
    type boolean;
    default true;
    description
        "Set prefix as a node
        representative prefix.";
}
leaf last-hop-behavior {
    type enumeration {
        enum explicit-null {
            description
                "Use explicit-null for the SID.";
        }
        enum no-php {
            description
                "Do no use PHP for the SID.";
        }
        enum php {
            description
                "Use PHP for the SID.";
        }
    }
    description
        "Configure last hop behavior.";
}

```

```

        description
            "List of prefix SID.";
    }
    description
        "This grouping defines cfg of prefix SID.";
}

/* Cfg */

augment "/rt:routing/rt:routing-instance" {
    description
        "This augments routing-instance
        configuration with segment-routing.";
    container segment-routing {
        leaf transport-type {
            type identityref {

```

```

        base segment-routing-transport;
    }
    default "segment-routing-transport-mpls";
    description "Dataplane to be used.";
}
container bindings {
    container mapping-server {
        if-feature mapping-server;
        container ipv4 {
            list mapping-entry {
                key prefix;

                leaf prefix {
                    type inet:ipv4-prefix;
                    description
                        "Base prefix used for mapping.";
                }
                leaf start-sid {
                    type uint32;
                    description
                        "Starting SID value to be associated
                        with prefix.";
                }
                leaf range {
                    type uint32;

```

```

        description
        "Describes how many SIDs could be
        allocated.";
    }
    description
    "Mapping entries.";
}
description
    "IPv4 mapping entries.";
}
container ipv6 {
    list mapping-entry {
        key prefix;

        leaf prefix {
            type inet:ipv6-prefix;
            description
            "Base prefix used for mapping.";
        }
        leaf start-sid {
            type uint32;
            description
            "Starting SID value to be associated

```

```

        with prefix.";
    }
    leaf range {
        type uint32;
        description
        "Describes how many SIDs could be
        allocated.";
    }
    description
    "Mapping entries.";
}
description
    "IPv6 mapping entries.";
}
description
    "Configuration of mapping-server
    local entries.";
}

```



```

        group-ID will be bundled together.
    ";
}
description
    "Control advertisement of S flag.
    Enable to advertise a common Adj-SID
    for parallel links.";
}
leaf advertise-protection {
    type enumeration {
        enum "single" {
            description
                "A single Adj-SID is associated
                with the adjacency and reflects
                the protection configuration.";
        }
        enum "dual" {
            description
                "Two Adj-SIDs will be associated
                with the adjacency if interface
                is protected. In this case
                one will be enforced with
                backup flag set, the other
                will be enforced to backup flag unset.
                In case, protection is not configured,
                a single Adj-SID will be advertised
                with backup flag unset.";
        }
    }
    description
        "If set, the Adj-SID refers to an
        adjacency being protected.";
}
description
    "Defines the adjacency SID properties.";
}
container prefix-sid {

```

```

container ipv4 {
    uses prefix-sid-cfg;
    description
        "Parameters associated with IPv4 prefix SID";
}

```



```

    }
    container ipv6 {
        uses prefix-sid-cfg;
        description
            "Parameters associated with IPv6 prefix SID";
    }
    description
        "Prefix SID configuration.";
    }
    description
        "List of interfaces.";
    }
    description
        "Interface configuration.";
    }
    description
        "segment routing global config.";
    }
}

```

```

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

    uses controlplane-cfg;
}

```

```

augment "/rt:routing/rt:routing-instance/rt:routing-protocols" +
    "/rt:routing-protocol/ospf:ospf/ospf:instance" {
    when "rt:type = 'ospf:ospfv2' or rt:type = 'ospf:ospfv3'" {
        description
            "This augment ISIS routing protocol when used";
    }
    description
        "This augments ISIS protocol configuration
        with segment routing.";
}

```

```
    uses controlplane-cfg;
}

/* Operational states */

augment "/rt:routing-state/rt:routing-instance" {
  description
    "This augments the operational states
    with segment-routing.";
  container segment-routing {
    list label-blocks {
      leaf lower-bound {
        type uint32;
        description
          "Lower bound of the label block.";
      }
      leaf upper-bound {
        type uint32;
        description
          "Upper bound of the label block.";
      }
      leaf size {
        type uint32;
        description
          "Number of indexes in the block.";
      }
      leaf free {
        type uint32;
        description
          "Number of indexes free in the block.";
      }
      leaf used {
        type uint32;
        description
          "Number of indexes used in the block.";
      }
      description
        "List of labels blocks currently
        in use.";
    }
  }
  container global-sid-list {
    list sid {
      key "target sid source source-protocol binding-type";
      ordered-by system;
    }
  }
}
```

leaf target {

```
    type string;
    description
      "Defines the target of the binding.
      It can be a prefix or something else.";
  }
  leaf sid {
    type uint32;
    description
      "Index associated with the prefix.";
  }
  leaf algorithm {
    type uint8;
    description
      "Algorithm to be used for the prefix
      SID.";
  }
  leaf source {
    type inet:ip-address;
    description
      "IP address of the router than own
      the binding.";
  }
  leaf used {
    type boolean;
    description
      "Defines if the binding is used
      in forwarding plane.";
  }
  leaf source-protocol {
    type leafref {
      path "/rt:routing-state/rt:routing-instance/" +
        "rt:routing-protocols/rt:routing-protocol/rt:name";
    }
    description
      "Rtg protocol that owns the binding";
  }
  leaf binding-type {
    type enumeration {
      enum prefix-sid {
        description
```

```

        "Binding is learned from
        a prefix SID.";
    }
    enum binding-tlv {
        description
            "Binding is learned from
            a binding TLV.";
    }

```

```

    }
    description
        "Type of binding.";
    }
    description
        "Binding.";
    }
    description
        "List of prefix and SID associations.";
    }
    description
        "Segment routing operational states.";
    }
}

/* Notifications */

notification segment-routing-global-sid-collision {
    leaf received-target {
        type string;
        description
            "Target received in the controlplane that
            caused SID collision.";
    }
    leaf original-target {
        type string;
        description
            "Target already available in database that have the same SID
            as the received target.";
    }
    leaf index {

```

```

    type uint32;
    description
      "Value of the index used by two different prefixes.";
  }
  leaf routing-protocol {
    type leafref {
      path "/rt:routing-state/rt:routing-instance/" +
        "rt:routing-protocols/rt:routing-protocol/rt:name";
    }
    description
      "Routing protocol reference that received the event.";
  }
  description
    "This notification is sent when a new mapping is learned
    , containing mapping

```

```

    where the SID is already used.
    The notification generation must be throttled with at least
      a 5 second gap. ";
  }
  notification segment-routing-index-out-of-range {
    leaf received-target {
      type string;
      description
        "Target received in the controlplane
        that caused SID collision.";
    }
    leaf received-index {
      type uint32;
      description
        "Value of the index received.";
    }
  }
  leaf routing-protocol {
    type leafref {
      path "/rt:routing-state/rt:routing-instance/" +
        "rt:routing-protocols/rt:routing-protocol/rt:name";
    }
    description
      "Routing protocol reference that received the event.";
  }
  description
    "This notification is sent when a binding

```

```
        is received, containing a segment index
        which is out of the local configured ranges.
        The notification generation must be throttled with at least
        a 5 second gap. ";
    }

}

<CODE ENDS>
```

## [8.](#) Security Considerations

TBD.

## [9.](#) Acknowledgements

TBD.

---

Litkowski, et al. Expires September 6, 2015 [Page 19]

---

Internet-Draft sr-yang-cfg March 2015

## [10.](#) IANA Considerations

TBD.

## [11.](#) Normative References

[I-D.ietf-isis-segment-routing-extensions]  
Previdi, S., Filsfils, C., Bashandy, A., Gredler, H.,  
Litkowski, S., Decraene, B., and J. Tantsura, "IS-IS  
Extensions for Segment Routing", [draft-ietf-isis-segment-  
routing-extensions-03](#) (work in progress), October 2014.

[I-D.ietf-ospf-segment-routing-extensions]  
Psenak, P., Previdi, S., Filsfils, C., Gredler, H.,  
Shakir, R., Henderickx, W., and J. Tantsura, "OSPF  
Extensions for Segment Routing", [draft-ietf-ospf-segment-  
routing-extensions-04](#) (work in progress), February 2015.

[I-D.ietf-spring-segment-routing]

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

[I-D.psenak-ospf-segment-routing-ospfv3-extension]

Psenak, P., Previdi, S., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPFv3 Extensions for Segment Routing", [draft-psenak-ospf-segment-routing-ospfv3-extension-02](#) (work in progress), July 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.

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

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", [RFC 6242](#), June 2011.

Litkowski, et al.

Expires September 6, 2015

[Page 20]

---

Internet-Draft

sr-yang-cfg

March 2015

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", [RFC 6536](#), March 2012.

#### Authors' Addresses

Stephane Litkowski  
Orange Business Service

Email: [stephane.litkowski@orange.com](mailto:stephane.litkowski@orange.com)

Acee Lindem  
Cisco Systems

Email: [acee@cisco.com](mailto:acee@cisco.com)

Pushpasis Sarkar  
Juniper Networks

Email: [psarkar@juniper.net](mailto:psarkar@juniper.net)

Ing-Wher Chen  
Ericsson

Email: [ing-wher.chen@ericsson.com](mailto:ing-wher.chen@ericsson.com)