

I2RS
Internet-Draft
Intended status: Standards Track
Expires: July 6, 2016

S. Hares
L. Dunbar
Huawei
January 3, 2016

A Yang model for I2RS service topology
draft-hares-i2rs-service-topo-dm-01.txt

Abstract

This document defines I2RS protocol-independent service layer virtual topology data model. This data model utilizes the concepts in the generic I2RS topology model of virtual networks (node, links, termination points) and cross-layer topologies. This virtual service topology may be a composite layer created from the combination of protocol-dependent service layers. Protocol-dependent services layers include: L3VPN, L2VPN, EVPN, E-Tree, and others.

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 July 6, 2016.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 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

1.	Introduction	2
1.1.	Conventions used in this document	2
1.2.	Base Model: the Service-Topology Component	2
2.	High level Yang architecture	3
2.1.	Network level	3
2.2.	Node level	4
2.3.	Service Link and Termination point	4
3.	Yang Data Model	5
4.	IANA Considerations	19
5.	Security Considerations	19
6.	References	19
6.1.	Normative References	19
6.2.	Informative References	19
	Authors' Addresses	19

1. Introduction

Service topology in [I-D.ietf-i2rs-yang-network-topo] includes the a virtual topology for a service layer above the L1, L2, and L3 layers. This virtual topology has the generic topology elements of node, link, and terminating point. The virtual service topology is a network-wide topology stored on one routing system which an I2RS agent is connected to.

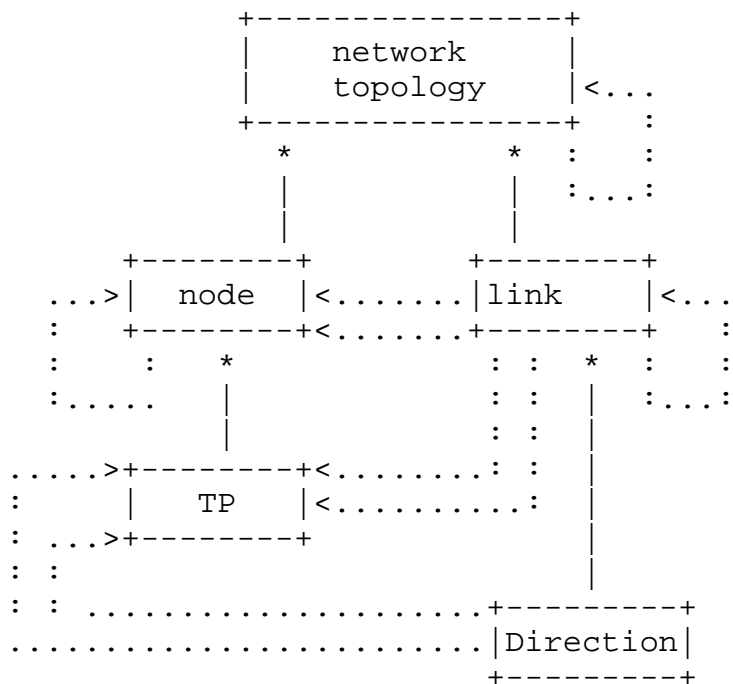
The virtual service topology is a composite of the available services topologies IETF has standard virtual network topologies for L3VPN, L2VPN, and EVPN services, E-TREE services, Seamless MPLS topologies within an As and others. This yang module provides the composite protocol independent service topology that these protocol dependent topologies plug into

1.1. Conventions used in this document

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 [RFC2119].

1.2. Base Model: the Service-Topology Component

The following diagram contains an informal graphical depiction of the main elements of the information model:



2. High level Yang architecture

This section describes the Yang High level architecture.

2.1. Network level

The service topology network level defines the following high-level yang architecture:

```

module: i2rs-service-topologies
  augment /nw:network/nw:network-types:
    +--rw service-topologies-types
  augment /nw:network:
    +--rw service-topology-attributes
      +--rw name? string
      +--rw flag flag_type;
      +--rw composite-flag identity-ref
      +--rw service-topo-id network-id
      +--rw node-count uint32
      +--rw composite-flag_status identity-ref
  
```

The service topology attributes for a network include the following

name - name of the service topology,

flag - generic topology flag,

composite-flags - bit mask with flags of service layer topologies network topology node available to create service topology from. These topologies include: L3VPN, L2VPN, and EVPN services, E-TREE services, Seamless MPLS topologies within an AS and others.

service-topo-id - service topology identifier.

node-count - count of nodes in composite service topology,

composite-flag_status - status flag for each of the composite topologies on whether portions of the topology are included in the composite.

2.2. Node level

```
module: i2rs-service-topologies
```

```
....
```

```
augment /nw:network/nw:node
```

```
  +--rw node-service-attributes
```

```
    +--rw name?          inet:domain-name
```

```
    +--rw composite_flag* identityref;
```

```
      +--rw service-node-id uint32
```

```
      +--rw node-svc_status* identityref;
```

The additional fields in the service attributes are the following:

name - name of network node,

flag - generic topology node flag,

service-node-id - the id for the service node in the composite network,

node-svc-type - the type of service node. The service node can be a member of one of the existing topology type (L3VPN, L2VPN, EVPN, E-TREE, Seamless MPLS, MPLS-TE, MPLS node, or I2RS created).

2.3. Service Link and Termination point

```

augment /nw:network/nt:link:
  +--rw service-link-attributes
    +--rw name?      string
    +--rw svc-link-type identityref
    +--rw metric?   uint32
augment /nw:network/nw:node/nt:termination-point:
  +--rw service-termination-point-attributes
    +--rw tp-svc-id
  +--rw (supporting-termination-point) ()
    +--:(service)
      | +--rw service-network-id  leafref
      | +--rw service-node-id     leafref
      | +--rw service-tp-id       leafref
      | +--:(ip)
      | +--rw ip-address          inet:ip-address
    +--:(unnumbered)
      +--rw unnumbered-id?      uint32

```

The augmentation to the service topology is the service link attributes which include:

name - name of the link,

svc-link-type - the service link type used to create this composite service link.

metric - the metric of the service type

The augmentation to the termination point include the following

tp-svc-id - service id for the termination point,

supporting termination point* - with a) references to the service network id, node id and termination point for service id, or b) an ip address, or c) an unnumbered link address

3. Yang Data Model

```

<CODE BEGINS> file "ietf-i2rs-service-topology@2016-01-03.yang"
module ietf-i2rs-service-topology{
  namespace "urn:ietf:params:xml:ns:yang:ietf-i2rs-service-topology";
  prefix i2rs-st;

  import ietf-inet-types {
    prefix inet;
  }

  import ietf-network {

```

```
    prefix nw;
  }
import ietf-network-topology {
  prefix "nt";
}

organization "IETF";
contact
  "email: shares@ndzh.com;
  email: linda.dunbar@huawei.com;
  ";

description
  "This module defines a model
  for the service topology.";

revision 2016-01-03{
  description
    "Version 1 - initial version;
    Version 2 - yang format fixed";

  reference "draft-hares-i2rs-service-topo-dm-01.txt";
}

identity svc-topo-flag-identity {
  description "Base type for svc flags";
}
identity l3vpn-svc-topo {
  base svc-topo-flag-identity;
  description "L3VPN service type";
}
identity l2vpn-svc-topo {
  base svc-topo-flag-identity;
  description "L2VPN service type";
}
identity EVPN-svc-topo {
  base svc-topo-flag-identity;
  description "EVPN service type";
}
identity Seamless-MPLS-svc-topo {
  base svc-topo-flag-identity;
  description "Seamless MPLS service type";
}
identity Etree-svc-topo {
  base svc-topo-flag-identity;
  description "Seamless MPLS service type";
}
identity I2rs-svc-topo {
```

```
    base svc-topo-flag-identity;
    description "I2RS create service topo";
}

identity svc-tp-type {
    description "Base type for service
        termination-point type flags";
}
identity svc-tp-type-service {
    base svc-tp-type;
    description "service type";
}
identity svc-tp-type-ip {
    base svc-tp-type;
    description "service IP";
}
identity svc-tp-type-unnum {
    base svc-tp-type;
    description "service unnumbered link";
}

    identity service-topology-types{
        description
            "service topology type";
    }

grouping service-topology-types {
    leaf service-type {
        type identityref {
            base svc-topo-flag-identity;
        }
        description "list of service
            topology type supported";
    }
    description
        "service topology types";
}

grouping service-topology-attributes {
    leaf name {
        type string;
        description "name of service
            topology";
    }
    leaf composite-flag {
        type identityref {
            base service-topology-types;
        }
    }
}
```

```

        description "other topologies
        this topology is configured to
        be a composite of
        (L3VPN, L2VPN, I2RS only)";
    }
    leaf service-topo-id {
        type nw:network-id;
        description "service topology id
        to a service topology instance.";
    }
    leaf service-id-number {
        type uint32;
        description "ID for topology";
    }
    leaf node-count {
        type uint32;
        description "count of service level nodes
        in the network.";
    }
    leaf composite-flag-status {
        type identityref {
            base svc-topo-flag-identity;
        }
        description "other topologies
        this topology is currently a
        composite of
        (L3VPN, L2VPN, I2RS only)";
    }
    description "Group of attributes for
    service topology";
}

grouping node-svc-attribute {
    leaf domain-name {
        type inet:domain-name;
        description "Domain name for node";
    }
    leaf composite-flag {
        type identityref {
            base svc-topo-flag-identity;
        }
        description "virtual network
        node can be composite of the
        topologies list
        (L3VPN, L2VPN, I2RS only)";
    }
    leaf service-node-id {
        type uint32;
        description "ID for node at

```



```

        service level";
    }
    leaf node-svc_status {
        type identityref {
            base service-topology-types;
        }
        description "other topologies
            this topology is currewntly
            be composed of
                (L3VPN, L2VPN, I2RS only)";
    }
    description
    "grouping of composite flag";
}

grouping service-link-attributes {
    leaf name {
        type string;
        description "name of
            service link";
    }
    leaf link-id {
        type uint32;
        description "link id";
    }
    leaf svc-link-type {
        type identityref {
            base service-topology-types;
        }
        description "other topologies
            this link is current a
                composite of
                (L3VPN, L2VPN, I2RS only)";
    }
    leaf metric {
        type uint32;
        description "link metric
            which may need to expand or
                link to TE topologies.";
    }
    description "grouping of
        service link attribute";
}

grouping service-termination-point-attributes {
    leaf svc-tp-id {
        type uint32;
    }
}

```

```

    description "termination point id";
  }
  container supporting-termination-point {
    leaf svc-tp-type {
      type identityref {
        base svc-tp-type;
      }
      description "other topologies
        this link termination point is
        part of (L3VPN, L2VPN,
          or I2RS only)";
    }
    choice svc-tp-support-type {
      case svc-tp-type-service {

        leaf service-network-id {
          type uint32;
          description "service network id";
        }
        leaf service-node-id {
          type uint32;
          description "service node id";
        }
        leaf service-link-id {
          type uint32;
          description "service link id";
        }
        description "network, node,
          tp that supports this
          termination point";
      }
      case svc-tp-type-inet {
        leaf ip-address {
          type inet:ip-address;
          description "ip address";
        }
        description "inet svc tp";
      }
    }

    case svc-tp-type-unnum {
      leaf unnumbered-id {
        type uint32;
        description "unnumbered id";
      }
      description "unnumber svc tp";
    }
    description "service termination
      point type cases";
  }

```

```

    }
    description "container of
        supporting termination point";
    }
        description
            "grouping of service-termination-point-attributes";
    }

/*
 * Data nodes
 */
    augment "/nw:networks/nw:network/nw:network-types" {
    uses service-topology-types;
        description
            "augment the network-tpyes with
            the service-topology-types grouping";
    }
augment "/nw:networks/nw:network/nw:node" {
    leaf name {
        type inet:domain-name;
        description "service name.";
    }
    list composite_fag {
        key "service-node-id";
        leaf service-node-id {
            type uint32;
            description "service node id.";
        }
        leaf node-svc-type {
            type string;
            description "node service type.";
        }
    }
    leaf-list next-hop {
        type uint32;
        description "next hop id.";
    }
    description
        "the list of composite flag.";
    }

    description "augments node list";
    }

augment "/nw:networks/nw:network" {
    uses service-topology-attributes;

```

```

        description
        "augment the network with
        the servcie-topolgooy-attributes";
    }
    augment "/nw:networks/nw:network/nw:node" {
        uses node-svc-attribute;
        description
        "augment the node with the node-svc-attribute";
    }
    augment "/nw:networks/nw:network/nt:link" {
        uses service-link-attributes;
        description
        "augment the link with
        service-link-attributes";
    }
    augment "/nw:networks/nw:network/nw:node/nt:termination-point" {
        uses service-termination-point-attributes;
        description
        "augment the termination-point with
        service-termination-point-attributes";
    }
} // module i2rs-service-topology
<CODE ENDS>

```

```

<CODE BEGINS> file "ietf-i2rs-service-topology@2016-01-03.yang"
    module ietf-i2rs-service-topology {
        namespace "urn:ietf:params:xml:ns:yang:ietf-i2rs-service-topology";
        prefix i2rs-st;

        import ietf-network {
            prefix nw;
        }
        import ietf-network-topology {
            prefix "nt";
        }

        import ietf-inet-types {
            prefix inet;
        }

        organization "TBD";
        contact "TBD";
        description
            "This module defines a model for the service topology.";

        revision 2016-01-03 {

```

```
description
  "Version 1 - initial version;
   Version 2 - yang format fixed;"
reference "draft-draft-ietf-i2rs-yang-network-topo-01.txt";
}

identity svc-topo-flag-identity {
  description "Base type for svc flags";
}
identity l3vpn-svc-topo {
  base svc-topo-flag-identity;
  description "L3VPN service type";
}
identity l2vpn-svc-topo {
  base svc-topo-flag-identity;
  description "L2VPN service type";
}
identity EVPN-svc-topo {
  base svc-topo-flag-identity;
  description "EVPN service type";
}
identity Seamless-MPLS-svc-topo {
  base svc-topo-flag-identity;
  description "Seamless MPLS service type";
}
identity Etree-svc-topo {
  base svc-topo-flag-identity;
  description "Seamless MPLS service type";
}
identity I2rs-svc-topo {
  base svc-topo-flag-identity;
  description "I2RS create service topo";
}
identity svc-tp-type {
  description "Base type for service
termination-point type flags";
}
identity svc-tp-type-service {
  base svc-tp-type;
  description "service type";
}
identity svc-tp-type-ip {
  base svc-tp-type;
  description "service IP";
}
identity svc-tp-type-unnum {
  base svc-tp-type;
  description "service unnumbered link";
}
```

```
    }

    identity service-topology-types{
      description
        "service topology type";
    }

grouping service-topology-types {
  leaf service-type {
    type identityref {
      base svc-topo-flag-identity;
    }
    description
      "list of topology type
      supported";
  }
  description
    "service topology types";
}
grouping service-topology-attributes {
  description
    "Grouping for the attribute for the
    service topology";
  leaf name {
    type string;
  }
  description
    "name of service
    topology";
  leaf composite-flag {
    type identityref {
      base service-topology-types;
    }
    description
      "other topologies
      this topology is configured to
      be a composite of
      (L3VPN, L2VPN, I2RS only)";
  }
  leaf service-topo-id {
    type leafref {
      path "/nw:network/nw:network-id";
    }
    description "An absolute
      reference to a network topology instance.";
  }
  leaf service-id-number {
    type uint32;
    description "ID for topology";
  }
}
```

```

        leaf node-node {
            type uint32;
            description "count of service level nodes
                in the network.";
        }
        leaf composite-flag-status {
            type identityref {
                base svc-topo-flag-identity;}
            description "other topologies
                this topology is currently a
composite of
                (L3VPN, L2VPN, I2RS only)";
        }
    }
    grouping node-svc-attribute {
        leaf domain-name{
            type inet:domain-name;
            description "Domain name for node";
        }
        leaf composite-flag {
            type identityref {
                base svc-topo-flag-identity;}
            description "virtual network
                node can be composite of the
                topologies list
                (L3VPN, L2VPN, I2RS only)";
        }
        leaf service-node-id {
            type uint32;
            description "ID for node at
                service level";
        }
        leaf node-svc_status {
            type identityref {
                base service-topology-types;}
            description "other topologies
                this topology is currewntly
be a composite of
                (L3VPN, L2VPN, I2RS only)";
        }
        description
            "grouping of composite flag";
    }
    grouping service-link-attributes {
        leaf name
            {type string;
            description
                "name of

```

```

        service link";}
    leaf link-id
        {type uint32;
        description
        "link id";}
    leaf svc-link-type {
        type identityref {
            base service-topology-types;}
        description "other topologies
        this link is current a composite of
        (L3VPN, L2VPN, I2RS only)";
    }
    leaf metric {
        type uint32;
        description "link metric";
    // may need to expand with TE topologies
    }
    description
    "grouping of service link attribute";
}
grouping service-termination-point-attributes {
    leaf svc-tp-id {
        type uint32;
        description "termination point id";
    }
    container supporting-termination-point {
        leaf svc-tp-type {
            type identityref {
                base svc-tp-type;
            }
            description "other topologies
            this link termination point is
            part of (L3VPN, L2VPN,
                I2RS only)";}
        choice svc-tp-support-type{
            case svc-tp-type-service {
                description "network, node,
                tp that supports this
                termination point";
                leaf service-network-id {
                    type uint32;
                    description
                        "service network id";
                }
            }
            leaf service-node-id {
                type uint32;
                description
                    "service node id";
            }
        }
    }
}

```



```

    }
    leaf service-link-id {
        type uint32;
        description
            "service link id";
    }
}
case svc-tp-type-inet {
    leaf ip-address{
        type inet:ip-address;
        description
            "ip address";
    }
}
case svc-tp-type-unnum {
    leaf unnumbered-id {
        type uint32;
        description
            "unnumbered id";
    }
}
description
    "service termination
    point type";
}
description
    "container of supporting termination point";
}
description
    "grouping of service-termination-point-attributes";
}

augment "/nw:network/nw:node"{
    leaf name{
        type inet:domain-name;
        description
            "service name.";
    }
    list composite_fag{
        key "service-node-id";
        leaf service-node-id{
            type uint32;
            description
                "service node id.";
        }
    }
    leaf node-svc-type{
        type string;
    }
}

```

```

        description
        "node service type.";
    }
    leaf-list next-hop{
        type uint32;
        description
        "next hop id.";
    }
    description
    "the list of composite flag.";
}

description
"augment the node list";
}

augment "/nw:network/nw:network-types"{
    uses service-topology-types;
    description
    "augment the network-tpyes with
    the servcie-topology-types grouping";
}
augment "/nw:network"{
    uses service-topology-attributes;
    description
    "augment the network with
    the servcie-topolgooy-attributes";
}
augment "/nw:network/nw:node"{
    uses node-svc-attribute;
    description
    "augment the node with the node-svc-attribute";
}
augment "/nw:network/nt:link"{
    uses service-link-attributes;
    description
    "augment the link with
    service-link-attributes";
}
augment "/nw:network/nw:node/nt:termination-point"{
    uses service-termination-point-attributes;
    description
    "augment the termination-point with
    service-termination-point-attributes";
}
} // module i2rs-service-topology
<CODE ENDS>

```

4. IANA Considerations

TBD

5. Security Considerations

TBD

6. References

6.1. Normative References

[I-D.ietf-i2rs-yang-network-topo]

Clemm, A., Medved, J., Varga, R., Tkacik, T., Bahadur, N., and H. Ananthakrishnan, "A Data Model for Network Topologies", draft-ietf-i2rs-yang-network-topo-02 (work in progress), December 2015.

[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>>.

6.2. Informative References

[I-D.ietf-i2rs-yang-l3-topology]

Clemm, A., Medved, J., Varga, R., Tkacik, T., Liu, X., Bryskin, I., Guo, A., Ananthakrishnan, H., Bahadur, N., and V. Beeram, "A YANG Data Model for Layer 3 Topologies", draft-ietf-i2rs-yang-l3-topology-01 (work in progress), December 2015.

Authors' Addresses

Susan Hares
Huawei
7453 Hickory Hill
Saline, MI 48176
USA

Email: shares@ndzh.com

Linda Dunbar
Huawei
USA

Email: linda.dunbar@huawei.com