I2RS Working Group
Internet-Draft
Intended status: Standards Track
Expires: June 25, 2018

Y. Zhuang
D. Shi
Huawei
R. Gu
China Mobile
H. Ananthakrishnan
Packet Design

December 22, 2017

A YANG Data Model for Fabric Topology in Data Center Network
draft-ietf-i2rs-yang-dc-fabric-network-topology-03

Abstract

This document defines a YANG data model for fabric topology in Data Center Network.

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 https://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 June 25, 2018.

Table of Contents

1. Introduction

Normally, a data center network is composed of single or multiple fabrics which are also known as PODs (a Point Of Delivery). These fabrics may be heterogeneous due to implementation of different technologies while DC network upgrading or enrolling new techniques and features. For example, Fabric A may use VXLAN while Fabric B may use VLAN within a DC network. Likewise, a legacy Fabric may use VXLAN while a new Fabric B implemented technique discussed in NVO3 WG such as GPE[I-D. draft-ietf-nvo3-vxlan-gpe] may be built due to DC expansion and upgrading. The configuration and management of such DC networks with heterogeneous fabrics will be sophisticated and complex.

Luckily, for a DC network, a fabric can be considered as an atomic structure to provide network services and management, as well as expand network capacity. From this point of view, the miscellaneous DC network management can be decomposed to task of managing each fabric respectively along with their connections, which can make the entire management much concentrated and flexible, also easy to expand.

With this purpose, this document defines a YANG data model for the Fabric-based Data center topology by using YANG [6020][7950]. To do so, it augments the generic network and network topology data models defined in [I-D.ietf-i2rs-yang-network-topo] with information specific to Data Center fabric network.

This model defines the generic configuration and operational state for a fabric-based network topology, which can be extended by vendors with specific information. This model can then be used by a network controller to represent its view of the fabric topology that it controls and expose it to network administrators or applications for DC network management.

With the context of topology architecture defined in [I-D.ietf-i2rs-yang-network-topo] and [I.D. draft-ietf-i2rs-usecase-reqs-summary], this model can also be treated as an application of I2RS network topology model [I-D.ietf-i2rs-yang-network-topo] in the scenario of Data center network management. It can also act as a service topology when mapping network elements at fabric layer to elements to other topologies, such as L3 topology defined in [I.D. draft-ietf-i2rs-yang-l3-topology-01.

By using this fabric topology model, people can treat a fabric as an entity and focus on characteristics of fabrics (such as encapsulation type, gateway type, etc.) as well as their interconnections while putting the underlay topology aside. As such, clients can consume the topology information at fabric level, while no need to be aware of entire set of links and nodes in underlay networks. The configuration of a fabric topology can be made by a network administractor to the controller by adding physical devices and links of a fabric into a fabric network. Alternatively, the fabric topology can also learnt from the underlay network infrastructure.

2. Definitions an Acronyms

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].

2.1. Terminology

DC Fabric: also known as POD, is a module of network, compute, storage, and application components that work together to deliver networking services. It is a repeatable design pattern, and its components maximize the modularity, scalability, and manageability of data centers.

2.2. Tree diagram

The following notations are used within the data tree and carry the meaning as below.

```
Each node is printed as:
  <status> <flags> <name> <opts> <type>

  <status> is one of:
       +  for current
       x  for deprecated
       o  for obsolete
  <flags> is one of:
      rw for configuration data
      ro for non-configuration data
      -x for rpcs
      -n for notifications
  <name> is the name of the node

  If the node is augmented into the tree from another module, its name
  is printed as <prefix>:<name>.
  <opts> is one of:
       ?  for an optional leaf or choice
       !  for a presence container
       *  for a leaf-list or list
       [<keys>] for a list's keys
  <type> is the name of the type for leafs and leaf-lists
```

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

3. Model Overview

This section provides an overview of the DC Fabric topology model and its relationship with other topology models.

3.1. Topology Model structure

The relationship of the DC fabric topology model and other topology models is shown in the following figure (dotted lines in the figure denote augmentations).

```
            +------------------------+
            |      network model     |
            +------------------------+
                         |
                         |
            +------------V-----------+
            | network topology model |
            +------------------------+
                         |
        +-----------+-----+------+-------------+
        |           |            |             |
    +---V----+  +---V----+   +---V----+   +----V---+
    |   L1   |  |   L2   |   |   L3   |   | Fabric |
    |topology|  |topology|   |topology|   |topology|
    |  model |  |  model |   |  model |   |  model |
    +--------+  +--------+   +--------+   +--------+
```

From the perspective of resource management and service provisioning for a Data Center network, the fabric topology model augments the basic network topology model with definitions and features specific to a DC fabric, to provide common configuration and operations for heterogeneous fabrics.

3.2. Fabric Topology Model

The fabric topology model module is designed to be generic and can be applied to data center fabrics built with different technologies, such as VLAN, VXLAN etc. The main purpose of this module is to configure and manage fabrics and their connections. provide a fabric-based topology view for data center network applications.

3.2.1. Fabric Topology

In the fabric topology module, a fabric is modeled as a node in the network, while the fabric-based Data center network consists of a set of fabric nodes and their connections known as "fabric port". The following is the snip of the definition to show the main structure of the model:

```
module: ietf-fabric-topology
augment /nw:networks/nw:network/nw:network-types:
   +--rw fabric-network!
augment /nw:networks/nw:network/nw:node:
   +--rw fabric-attributes
          +--rw fabric-id?              fabric-id
          +--rw name?           string
          +--rw type?           fabrictype:underlay-network-type
          +--rw description?    string
          +--rw options
          +--...
augment /nw:networks/nw:network/nw:node/nt:termination-point:
   +--ro fport-attributes
          +--ro name?          string
          +--ro role?          fabric-port-role
          +--ro type?          fabric-port-type
```

The fabric topology module augments the generic ietf-network and ietf-network-topology modules as follows:

- o A new topology type "ietf-fabric-topology" is introduced and added under the "network-types" container of the ietf-network module.

- o Fabric is defined as a node under the network/node container. A new container of "fabric-attributes" is defined to carry attributes for a fabric network such as gateway mode, fabric types, involved device nodes and links etc.

- o Termination points (in network topology module) are augmented with fabric port attributes defined in a container. The "termination-point" here can represent the "port" of a fabric that provides connections to other nodes, such as device internally, another fabric externally and also end hosts.

Details of fabric node and fabric termination point extension will be explained in the following sections.

3.2.2. Fabric node extension

As a network, a fabric itself is composed of set of network elements i.e. devices, and related links. As stated previously, the configuration of a fabric is contained under the "fabric-attributes" container depicted as follows:

```
                +--rw fabric-attributes
                   +--rw fabric-id?      fabric-id
                   +--rw name?           string
                   +--rw type?           fabrictype:underlay-network-type
                   +--rw vni-capacity
                   |  +--rw min?   int32
                   |  +--rw max?   int32
                   +--rw description?    string
                   +--rw options
                   |  +--rw gateway-mode?           enumeration
                   |  +--rw traffic-behavior?       enumeration
                   |  +--rw capability-supported*   fabrictype:service-
capabilities
                   +--rw device-nodes* [device-ref]
                   |  +--rw device-ref    fabrictype:node-ref
                   |  +--rw role?         fabrictype:device-role
                   +--rw device-links* [link-ref]
                   |  +--rw link-ref    fabrictype:link-ref
                   +--rw device-ports* [port-ref]
                          +--rw port-ref     fabrictype:tp-ref
                          +--rw port-type?   fabrictypes:port-type
                          +--rw bandwidth?   fabrictypes:bandwidth
```

As in the module, additional data objects for nodes are introduced by augmenting the "node" list of the network module. New objects include fabric name, type of the fabric, descriptions of the fabric as well as a set of options defined in an "options" container. The options container includes type of the gateway-mode (centralized or distributed) and traffic-behavior (whether acl needed for the traffic).

Also, it defines a list of device-nodes and related links as supporting-nodes to form a fabric network. These device nodes and links are leaf-ref of existing nodes and links in the underlay topology. For the device-node, the "role" object is defined to represents the role of the device within the fabric, such as "SPINE" or "LEAF", which should work together with gateway-mode.

3.2.3. Fabric termination-point extension

Since the fabric can be considered as a node, in this concept, "termination-points" can represent "ports" of a fabric that connects to other fabrics or end hosts, besides representing ports that connect devices inside the fabric itself.

As such, the "termination-point" in the fabric topology has three roles, that are internal TP that connects to devices within a fabric,

external TP that connects to outside network, and access TP to end hosts.

The set of "termination-point" of a fabric indicates all connections of the fabric, including its internal connections, interconnections with other fabrics and also connections to end hosts for a DC network.

The structure of fabric ports is as follows:

```
        augment /nw:networks/nw:network/nw:node/nt:termination-point:
              +--ro fport-attributes
                 +--ro name?          string
                 +--ro role?          fabric-port-role
                 +--ro type?          fabric-port-type
                 +--ro device-port?   tp-ref
                 +--ro (tunnel-option)?
```

It augments the termination points (in network topology module) with fabric port attributes defined in a container.

New nodes are defined for fabric ports which include name, role of the port within the fabric (internal port, external port to outside network, access port to end hosts), port type (l2 interface, l3 interface etc). By defining the device-port as a tp-ref, a fabric port can be mapped to a device node in the underlay network.

Also, a new container for tunnel-options is introduced to present the tunnel configuration on the port.

The termination points information are all learnt from the underlay networks but not configured by the fabric topology layer.

4. Fabric YANG Module

```
<CODE BEGINS> file "ietf-dc-fabric-types@2017-12-21.yang"
    module ietf-dc-fabric-types {

    yang-version 1.1;
    namespace "urn:ietf:params:xml:ns:yang:ietf-dc-fabric-types";
    prefix fabrictypes;

        organization
    "IETF I2RS (Interface to the Routing System) Working Group";

        contact
```

```
    "WG Web:    <http://tools.ietf.org/wg/i2rs/ >
     WG List:   <mailto:i2rs@ietf.org>

     Editor:    Yan Zhuang
                <mailto:zhuangyan.zhuang@huawei.com>

     Editor:    Danian Shi
                <mailto:shidanian@huawei.com>";

     description
        "This module contains a collection of YANG definitions for Fabric.
        Copyright (c) 2016 IETF Trust and the persons identified as
        authors of the code.  All rights reserved.

        Redistribution and use in source and binary forms, with or
        without modification, is permitted pursuant to, and subject
        to the license terms contained in, the Simplified BSD License
        set forth in Section 4.c of the IETF Trust's Legal Provisions
        Relating to IETF Documents
        (http://trustee.ietf.org/license-info).

        This version of this YANG module is part of
        draft-ietf-i2rs-yang-dc-fabric-network-topology;
        see the RFC itself for full legal notices.

        NOTE TO RFC EDITOR: Please replace above reference to
        draft-ietf-i2rs-yang-dc-fabric-network-topology-03 with RFC
        number when published (i.e. RFC xxxx).";

    revision "2017-12-21"{
        description
            "Initial revision.
             NOTE TO RFC EDITOR: Please replace the following reference to
             draft-ietf-i2rs-yang-dc-fabric-network-topology-03 with RFC
             number when published (i.e. RFC xxxx).";
        reference
            "draft-ietf-i2rs-yang-dc-fabric-network-topology-03";
        }

    identity fabric-type {
        description
           "Base type for fabric networks";
    }

    identity vxlan-fabric {
        base fabric-type;
        description "Vxlan fabric";
    }
```

```
    identity vlan-fabric {
        base fabric-type;
        description
           "Vlan fabric";
    }

    identity trill-fabric {
        base fabric-type;
        description "Trill fabric";
    }
    identity port-type {
        description
           "Base type for fabric port";
        }
    identity eth {
        base port-type;
        description "ETH";
    }
    identity serial {
        base port-type;
        description "Serial";
    }

    identity bandwidth {
        description "Base for bandwidth";
    }
    identity bw-1M {
        base bandwidth;
        description "1M";
    }
    identity bw-10M {
        base bandwidth;
        description "10M";
    }
    identity bw-100M {
        base bandwidth;
        description "100M";
    }
    identity bw-1G {
        base bandwidth;
        description "1G";
    }
    identity bw-10G {
        base bandwidth;
        description "10G";
    }
    identity bw-40G {
        base bandwidth;
```

```
        description "40G";
    }
    identity bw-100G{
        base bandwidth;
        description "100G";
    }

    identity device-role {
        description "Base for the device role in a fabric.";
    }
    identity spine {
        base device-role;
        description "This is a spine node in a fabric.";
    }
    identity leaf {
        base device-role;
        description "This is a leaf node in a fabric. ";
    }
    identity border {
        base device-role;
        description "This is a border node to connect to
        other fabric/network.";
    }
    identity fabric-port-role {
        description "Base for the port's role in a fabric.";
    }
    identity internal {
        base fabric-port-role;
        description "The port is used for devices to access
        each other within a fabric.";
    }
    identity external {
        base fabric-port-role;
        description "The port is used for a fabric to connect
        to outside network.";
    }
    identity access {
        base fabric-port-role;
        description "The port is used for an endpoint to
        connect to a fabric.";
    }

    /*
     * Typedefs
     */
    typedef service-capabilities {
        type enumeration {
            enum ip-mapping {
```

```
                description "NAT";
                        }
            enum acl-redirect{
                description "Acl redirect, which can provide
                SFC function";
                        }
            enum dynamic-route-exchange{
                description "Dynamic route exchange";
                        }
        }
        description
            "Capability of the device";
    }

    typedef port-type {
        type identityref {
             base port-type;
        }
        description "Port type: ethernet or serial or others.";
    }
    typedef bandwidth {
        type identityref {
             base bandwidth;
        }
        description "Bandwidth of the port.";
    }
    typedef node-ref {
        type instance-identifier;
        description "A reference to a node in topology";
    }

    typedef tp-ref {
        type instance-identifier;
        description "A reference to a termination point in topology";
    }

    typedef link-ref {
        type instance-identifier;
        description "A reference to a link in topology";
    }

    typedef underlay-network-type {
        type identityref {
             base fabric-type;
        }
        description "The type of physical network that implements this
        fabric.Examples are vlan, and trill.";
    }
```

```
    typedef device-role {
         type identityref {
             base device-role;
        }
        description "Role of the device node.";
    }
    typedef fabric-port-role {
        type identityref {
             base fabric-port-role;
        }
        description "Role of the port in a fabric.";
    }

    typedef fabric-port-type {
        type enumeration {
            enum layer2interface {
                description "L2 interface";
            }
            enum layer3interface {
                description "L3 interface";
            }
            enum layer2Tunnel {
                description "L2 tunnel";
            }
            enum layer3Tunnel {
                description "L3 tunnel";
            }
        }
                description
            "Fabric port type";
    }

    grouping fabric-port {
        description
            "Attributes of a fabric port.";
        leaf name {
            type string;
            description "Name of the port.";
        }
        leaf role {
            type fabric-port-role;
            description "Role of the port in a fabric.";
        }
        leaf type {
            type fabric-port-type;
            description "Type of the port";
        }
        leaf device-port {
```

```
            type tp-ref;
            description "The device port it mapped to.";
        }
        choice tunnel-option {
            description "Tunnel options to connect two fabrics.
            It could be L2 Tunnel or L3 Tunnel.";
        }
    }
}
<CODE ENDS>

<CODE BEGINS> file "ietf-dc-fabric-topology@2017-12-21.yang"
    module ietf-dc-fabric-topology {

    yang-version 1.1;
    namespace "urn:ietf:params:xml:ns:yang:ietf-dc-fabric-topology";
    prefix fabric;

    import ietf-network {
        prefix nw;

        reference
        "draft-ietf-i2rs-yang-network-topo-20
        NOTE TO RFC EDITOR:
        (1) Please replace above reference to
        draft-ietf-i2rs-yang-network-topo-20 with RFC
        number when published (i.e. RFC xxxx).
        (2) Please replace the date in the revision statement with the
         date of publication when published.";
    }

    import ietf-network-topology {
        prefix nt;

        reference
        "draft-ietf-i2rs-yang-network-topo-20
        NOTE TO RFC EDITOR:
        (1) Please replace above reference to
        draft-ietf-i2rs-yang-network-topo-20 with RFC
        number when published (i.e. RFC xxxx).
        (2) Please replace the date in the revision statement with the
         date of publication when published.";
    }

    import ietf-dc-fabric-types {
        prefix fabrictypes;

        reference
```

```
        "draft-ietf-i2rs-yang-dc-fabric-network-topology-03
        NOTE TO RFC EDITOR:
         (1) Please replace above reference to draft-ietf-i2rs-yang-dc
         -fabric-network-topology-03 with RFC number when publised
         (i.e. RFC xxxx).
         (2) Please replace the data in the revision statement with the
         data of publication when published.";
    }

    organization
    "IETF I2RS (Interface to the Routing System) Working Group";

    contact
    "WG Web:    <http://tools.ietf.org/wg/i2rs/ >
     WG List:   <mailto:i2rs@ietf.org>

     Editor:    Yan Zhuang
                <mailto:zhuangyan.zhuang@huawei.com>

     Editor:    Danian Shi
                <mailto:shidanian@huawei.com>"; 


    description
       "This module contains a collection of YANG definitions for Fabric.

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

        Redistribution and use in source and binary forms, with or
        without modification, is permitted pursuant to, and subject
        to the license terms contained in, the Simplified BSD License
        set forth in Section 4.c of the IETF Trust's Legal Provisions
        Relating to IETF Documents
        (http://trustee.ietf.org/license-info).

        This version of this YANG module is part of
        draft-ietf-i2rs-yang-dc-fabric-network-topology;see the RFC
        itself for full legal notices.

        NOTE TO RFC EDITOR: Please replace above reference to
        draft-ietf-i2rs-yang-dc-fabric-network-topology-03 with RFC
        number when published (i.e. RFC xxxx).";

        revision "2017-12-21"{
        description
             "Initial revision.
              NOTE TO RFC EDITOR: Please replace the following reference
```

```
              to draft-ietf-i2rs-yang-dc-fabric-network-topology-03 with
              RFC number when published (i.e. RFC xxxx).";
        reference
            "draft-ietf-i2rs-yang-dc-fabric-network-topology-03";
    }

    identity fabric-context {
        description
           "Identity of fabric context";
    }

    typedef fabric-id {
        type nw:node-id;
        description
            "An identifier for a fabric in a topology.
            The identifier is generated by compose-fabric RPC.";
    }

    //grouping statements
    grouping fabric-network-type {
        description "Identify the topology type to be fabric.";
        container fabric-network {
            presence "indicates fabric Network";
        description
          "The presence of the container node indicates fabric topology";
      }
    }

    grouping fabric-options {
        description "Options for a fabric";

        leaf gateway-mode {
            type enumeration {
                enum centralized {
                    description "The Fabric uses centerilized gateway, in
                    which gateway is deployed on SPINE node.";
                }
                enum distributed {
                    description "The Fabric uses distributed gateway, in
                    which gateway is deployed on LEAF node.";
                }
            }
            default "distributed";
            description "Gateway mode of the fabric";
        }

        leaf traffic-behavior {
            type enumeration {
```

```
                enum normal {
                    description "Normal";
                }
                enum policy-driven {
                    description "Policy driven";
                }
            }
            default "normal";
            description "Traffic behavior of the fabric";
        }

        leaf-list capability-supported {
            type fabrictypes:service-capabilities;
            description
               "Supported services of the fabric";
        }
    }

    grouping device-attributes {
        description "device attributes";
        leaf device-ref {
            type fabrictypes:node-ref;
            description
                "The device the fabric includes.";
        }
        leaf role {
            type fabrictypes:device-role;
            default fabrictypes:leaf;
            description
                "Role of the device node";
        }
    }

    grouping link-attributes {
                description "Link attributes";
        leaf link-ref {
            type fabrictypes:link-ref;
            description
                "The link it includes";
        }
    }

    grouping port-attributes {
        description "Port attributes";
        leaf port-ref {
            type fabrictypes:tp-ref;
            description
                "The port it refers to.";
```

```
        }
        leaf port-type {
            type fabrictypes:port-type;
            description
                "Port type: ethernet or serial or others.";
        }
        leaf bandwidth {
            type fabrictypes:bandwidth;
            description
                "Bandwidth of the port.";
        }
    }

    grouping fabric-attributes {
        description "Attributes of a fabric";

        leaf fabric-id {
            type fabric-id;
            description
                "Fabric id";
        }

        leaf name {
            type string;
            description
                "Name of the fabric";
        }

        leaf type {
            type fabrictypes:underlay-network-type;
            description
                "The type of physical network that implements this
                 fabric.Examples are vlan, and trill.";
        }

        container vni-capacity {
            description "Number of vnis that the fabric has";
            leaf min {
                type int32;
                description
                    "Vni min capacity";
            }

            leaf max {
                type int32;
                description
                    "Vni max capacity";
            }
```

```
        }

        leaf description {
            type string;
            description
                "Description of the fabric";
        }

        container options {
            description "Options of the fabric";
            uses fabric-options;
        }

        list device-nodes {
            key device-ref;
            description "Device nodes that include in a fabric.";
            uses device-attributes;
        }

        list device-links {
            key link-ref;
            description "Links that include within a fabric.";
            uses link-attributes;
        }

        list device-ports {
            key port-ref;
            description "Ports that include in the fabric.";
            uses port-attributes;
        }

    }

        // augment statements

    augment "/nw:networks/nw:network/nw:network-types" {
    description
        "Introduce new network type for Fabric-based logical topology";

        uses fabric-network-type;
        }

    augment "/nw:networks/nw:network/nw:node" {
        when "/nw:networks/nw:network/nw:network-types/fabric:fabric-network" {
        description
          "Augmentation parameters apply only for networks
           with fabric topology";
      }
```

```
        description "Augmentation for fabric nodes created by faas.";

        container fabric-attributes {
                        description
                        "Attributes for a fabric network";

            uses fabric-attributes;
        }
    }

    augment "/nw:networks/nw:network/nw:node/nt:termination-point" {
        when "/nw:networks/nw:network/nw:network-types/fabric:fabric-network" {
        description
          "Augmentation parameters apply only for networks
           with fabric topology";
      }
        description "Augmentation for port on fabric.";

        container fport-attributes {
            config false;
            description
               "Attributes for fabric ports";
            uses fabrictypes:fabric-port;
        }
    }
}
<CODE ENDS>
```

5. IANA Considerations

This document registers the following namespace URIs in the "IETF XML Registry" [RFC3688]:

URI: urn:ietf:params:xml:ns:yang:ietf-dc-fabric-types Registrant Contact: The IESG. XML: N/A; the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-dc-fabric-topology Registrant Contact: The IESG. XML: N/A; the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-dc-fabric-topology-state Registrant Contact: The IESG. XML: N/A; the requested URI is an XML namespace.

This document registers the following YANG modules in the "YANG Module Names" registry [RFC6020]:

NOTE TO THE RFC EDITOR: In the list below, please replace references to "draft-ietf-i2rs-yang-dc-fabric-network-topology-03 (RFC form)" with RFC number when published (i.e. RFC xxxx).

Name: ietf-dc-fabric-types Namespace: urn:ietf:params:xml:ns:yang:ietf-dc-fabric-types Prefix: fabrictypes Reference: draft-ietf-i2rs-yang-dc-fabric-network-topology-03.txt (RFC form)

Name: ietf-dc-fabric-topology Namespace: urn:ietf:params:xml:ns:yang:ietf-dc-fabric-topology Prefix: fabric Reference: draft-ietf-i2rs-yang-dc-fabric-network-topology-03.txt (RFC form)

Name: ietf-dc-fabric-topology-state Namespace: urn:ietf:params:xml:ns:yang:ietf-dc-fabric-topology-state Prefix: sfabric Reference: draft-ietf-i2rs-yang-dc-fabric-network-topology-03.txt (RFC form)

6. Security Considerations

The YANG module defined in this document is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC5246].

The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.These are the subtrees and data nodes and their sensitivity/vulnerability in the ietf-dc-fabric-topology module:

fabric-attributes: A malicious client could attempt to sabotage the configuration of important fabric attributes, such as device-nodes, type.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability in the ietf-dc-fabric-topology module:

fport-attributes: A malicious client could attempt to read the connections of fabrics without permission, such as device-port, name.

7. Acknowledgements

We wish to acknowledge the helpful contributions, comments, and suggestions that were received from Alexander Clemm, Xufeng Liu, Susan Hares, Wei Song, Luis M. Contreras and Benoit Claise.

8. References

8.1. Normative References

[I-D.draft-ietf-i2rs-yang-l3-topology]
Clemm, A., Medved, J., Tkacik, T., Liu, X., Bryskin, I., Guo, A., Ananthakrishnan, H., Bahadur, N., and V. Beeram, "A YANG Data Model for Layer 3 Topologies", I-D draft-ietf-i2rs-yang-l3-topology-04, September 2016.

[I-D.draft-ietf-i2rs-yang-network-topo]
Clemm, A., Medved, J., Tkacik, T., Varga, R., Bahadur, N., and H. Ananthakrishnan, "A YANG Data Model for Network Topologies", I-D draft-ietf-i2rs-yang-network-topo-06, September 2016.

[I-D.draft-ietf-netmod-revised-datastores-06]
Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "A Revised Conceptual Model for YANG Datastores", I-D draft-ietf-netmod-revised-datastores-06, October 2017.

[I-D.draft-ietf-nvo3-vxlan-gpe]
Maino, F., Kreeger, L., and U. Elzur, "Generic Protocol Extension for VXLAN", I-D draft-ietf-i2rs-yang-network-topo-02, 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>.

[RFC5246] Dierks, T. and E. Rescorla, "Transport Layer Security (TLS) Protocol Version 1.2", August 2008, <http://www.rfc-editor.org/info/rfc5246>.

[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)", June 2011, <http://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", June 2011, <http://www.rfc-editor.org/info/rfc6242>.

[RFC6536] Bierman, A. and B. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", March 2012, <http://www.rfc-editor.org/info/rfc6536>.

[RFC6991] Schoenwaelder, J., "Common YANG Data Types", RFC 6991, July 2013.

[RFC7950] Bjorklund, M., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, Auguest 2016.

[RFC8040] Bierman, A., Bjorklund, B., and K. Watsen, "RESTCONF Protocol", Jan 2017, <http://www.rfc-editor.org/info/rfc8040>.

8.2. Informative References

[I-D.draft-ietf-i2rs-usecase-reqs-summary] Hares, S. and M. Chen, "Summary of I2RS Use Case Requirements", I-D draft-ietf-i2rs-usecase-reqs-summary-01, May 2015.

Appendix A. Non NMDA -state modules

The YANG module ietf-fabric-toplogy defined in this document augments two modules, ietf-network and ietf-network-topology, that are designed to be used in conjunction with implementations that support the Network Management Datastore Architecture (NMDA) defined in [I-D.draft-ietf-netmod-revised-datastores]. In order to allow implementations to use the model even in case when NMDA is not supported, a set of companion modules have been defined that represent a state model of networks and network topologies, ietf-network-state and ietf-network-topology-state, respectively.

In order to be able to use the model for fabric topologies defined in this in this document in conjunction with non-NMDA compliant implementations, a corresponding companion module needs to be introduced as well. This companion module, ietf-fabric-topology-state, mirrors ietf-fabric-topology. However, the module augments ietf-network-state (instead of ietf-network and ietf-network-topology) and all of its data nodes are non-configurable.

Like ietf-network-state and ietf-network-topology-state, ietf-fabric-topology-state SHOULD NOT be supported by implementations that support NMDA. It is for this reason that the module is defined in the Appendix.

The definition of the module follows below. As the structure of the module mirrors that of its underlying module, the YANG tree is not depicted separately.

```
<CODE BEGINS> file "ietf-dc-fabric-topology-state@2017-12-21.yang"
module ietf-dc-fabric-topology-state {

yang-version 1.1;
namespace "urn:ietf:params:xml:ns:yang:ietf-dc-fabric-topology-state";
prefix sfabric;

import ietf-network-state {
    prefix nws;
    reference
    "draft-ietf-i2rs-yang-network-topo-20
    NOTE TO RFC EDITOR:
    (1) Please replace above reference to
    draft-ietf-i2rs-yang-network-topo-20 with RFC
    number when published (i.e. RFC xxxx).
    (2) Please replace the date in the revision statement with the
     date of publication when published.";
}
import ietf-dc-fabric-types {
    prefix fabrictypes;

    reference
    "draft-ietf-i2rs-yang-dc-fabric-network-topology-03
    NOTE TO RFC EDITOR:
    (1) Please replace above reference to draft-ietf-i2rs-yang-dc
    -fabric-network-topology-03 with RFC number when published
    (i.e. RFC xxxx).
    (2) Please replace the data in the revision statement with the
    data of publication when published.";
}
import ietf-dc-fabric-topology {
    prefix fabric;

    reference
    "draft-ietf-i2rs-yang-dc-fabric-network-topology-03
     NOTE TO RFC EDITOR:
     (1) Please replace above reference to draft-ietf-i2rs-yang-dc
     -fabric-network-topology-03 with RFC number when published
     (i.e. RFC xxxx).
```

```
         (2) Please replace the data in the revision statement with the
         data of publication when published.";
    }

    organization
    "IETF I2RS (Interface to the Routing System) Working Group";

    contact
    "WG Web:    <http://tools.ietf.org/wg/i2rs/ >
     WG List:   <mailto:i2rs@ietf.org>

     Editor:    Yan Zhuang
                <mailto:zhuangyan.zhuang@huawei.com>

     Editor:    Danian Shi
                <mailto:shidanian@huawei.com>";


    description
        "This module contains a collection of YANG definitions
        for Fabric state, representing topology that is either
        learned, or topology that results from applying toplogy
        that has been configured per the ietf-dc-fabric-topology
        model, mirroring the corresponding data nodes in this model.

        This model mirrors the configuration tree of
        ietf-dc-fabric-topology, but contains only read-only state
        data. The model is not needed when the implemtation
        infrastructure supports the Network Management Datastore
        Architecture(NMDA).

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

        Redistribution and use in source and binary forms, with or
        without modification, is permitted pursuant to, and subject
        to the license terms contained in, the Simplified BSD License
        set forth in Section 4.c of the IETF Trust's Legal Provisions
        Relating to IETF Documents(http://trustee.ietf.org/license-info).

        This version of this YANG module is part of
        draft-ietf-i2rs-yang-dc-fabric-network-topology;see the RFC
        itself for full legal notices.

        NOTE TO RFC EDITOR: Please replace above reference to
        draft-ietf-i2rs-yang-dc-fabric-network-topology-03 with RFC number
        when published (i.e. RFC xxxx).";
```

```
    revision "2017-12-21" {
        description
            "Initial revision.
             NOTE TO RFC EDITOR: Please replace the following reference to
             draft-ietf-i2rs-yang-dc-fabric-network-topology-03 with RFC
             number when published (i.e. RFC xxxx).";
        reference
            "draft-ietf-i2rs-yang-dc-fabric-network-topology-03";
    }

    //grouping statements
    grouping fabric-network-type {
      description "Identify the topology type to be fabric.";
      container fabric-network {
         presence "indicates fabric Network";
         description
          "The presence of the container node indicates fabric Topology";
        }
    }

    grouping fabric-options {
      description "Options for a fabric";
      leaf gateway-mode {
        type enumeration {
          enum centralized {
            description "The Fabric uses centerilized gateway, in which
            gateway is deployed on SPINE node.";
          }
          enum distributed {
            description "The Fabric uses distributed gateway, in which
            gateway is deployed on LEAF node.";
          }
        }
        default "distributed";
        description "Gateway mode of the fabric";
      }

      leaf traffic-behavior {
        type enumeration {
          enum normal {
            description "Normal";
          }
          enum policy-driven {
            description "Policy driven";
          }
        }
        default "normal";
        description "Traffic behavior of the fabric";
```

```
      }

      leaf-list capability-supported {
        type fabrictypes:service-capabilities;
        description
          "Supported services of the fabric";
      }
    }

    grouping device-attributes {
      description "device attributes";
      leaf device-ref {
        type fabrictypes:node-ref;
        description "The device the fabric includes.";
      }
      leaf role {
        type fabrictypes:device-role;
        default fabrictypes:leaf;
        description "Role of the node";
      }
    }

    grouping link-attributes {
      description "Link attributes";
      leaf link-ref {
        type fabrictypes:link-ref;
        description "The link it includes";
      }
    }

    grouping port-attributes {
      description "Port attributes";
      leaf port-ref {
        type fabrictypes:tp-ref;
        description "The port it refers to.";
      }
      leaf port-type {
        type fabrictypes:port-type;
        description
          "Port type: ethernet or serial or others";
        }
      leaf bandwidth {
        type fabrictypes:bandwidth;
        description "Bandwidth of the port";
      }
    }

    grouping fabric-attributes {
```

```
      description "Attributes of a fabric";
      leaf fabric-id {
        type fabric:fabric-id;
        description "Fabric id";
      }
      leaf name {
        type string;
        description     "Name of the fabric";
      }
      leaf type {
        type fabrictypes:underlay-network-type;
         description
          "The type of physical network that implements this
          fabric.Examples are vlan, and trill.";
        }
      container vni-capacity {
        description "Number of vnis the fabric has";
          leaf min {
            type int32;
            description "Vni min capacity";
          }
          leaf max {
            type int32;
            description "Vni max capacity";
          }
      }
      leaf description {
        type string;
        description "Description of the fabric";
      }
      container options {
        description "Options of the fabric";
        uses fabric-options;
      }
      list device-nodes {
        key device-ref;
        description "Device nodes that include in a fabric.";
        uses device-attributes;
      }
      list device-links {
        key link-ref;
        description "Links that include within a fabric.";
        uses link-attributes;
      }
      list device-ports {
        key port-ref;
        description "Ports that include in the fabric.";
        uses port-attributes;
```

```
      }
    }

    // augment statements

    augment "/nws:networks/nws:network/nws:network-types" {
      description
        "Introduce new network type for Fabric-based logical topology";
      uses fabric-network-type;
    }

    augment "/nws:networks/nws:network/nws:node" {
      when "/nws:networks/nws:network/nws:network-types/sfabric:fabric-
network" {
        description "Augmentation parameters apply only for networks with
        fabric topology.";
      }
      description "Augmentation for fabric nodes.";
      container fabric-attributes-state {
        description
          "Attributes for a fabric network";
        uses fabric-attributes;
      }
    }
}
<CODE ENDS>
```

Authors' Addresses

Yan Zhuang
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: zhuangyan.zhuang@huawei.com

Danian Shi
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: shidanian@huawei.com

Rong Gu
China Mobile
32 Xuanwumen West Ave, Xicheng District
Beijing, Beijing 100053
China

Email: gurong_cmcc@outlook.com

Hariharan Ananthakrishnan
Packet Design

Email: hari@packetdesign.com