I2RS Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 28, 2018

Y. Zhuang
D. Shi
Huawei
R. Gu
China Mobile
H. Ananthakrishnan
Packet Design
March 27, 2018

A YANG Data Model for Fabric Topology in Data Center Networks
draft-ietf-i2rs-yang-dc-fabric-network-topology-08

Abstract

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

Table of Contents

1. Introduction

Normally, a data center (DC) network is composed of single or multiple fabrics which are also known as PODs (Points Of Delivery). These fabrics may be heterogeneous due to implementation of different technologies when a DC network is upgraded or new techniques and features are enrolled. For example, Fabric A may use VXLAN while Fabric B may use VLAN within a DC network. Likewise, an existing fabric may use VXLAN while a new fabric, for example a fabric introduced for DC upgrade and expansion, may implement a technique discussed in NVO3 WG, such as Geneve [I-D. draft-ietf-nvo3-geneve]. The configuration and management of such DC networks with heterogeneous fabrics will result in considerable complexity, requiring a fair amount of sophistication.

Luckily, for a DC network, a fabric can be considered as an atomic structure for management purposes. From this point of view, the management of the DC network can be decomposed into a set of tasks to manage each fabric separately, as well as the fabric interconnections. This way, the overall management task becomes very flexible and makes it easy to expand and adopt to DC networks that evolve over time.

As a basis for DC fabric management, this document defines a YANG data model [RFC6020][RFC7950] for fabric-based data center topology. To do so, it augments the generic network and network topology data models defined in [RFC8345] with information that is specific to Data Center fabric networks.

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

Within the context of topology architecture defined in [RFC8345] and [I.D. draft-ietf-i2rs-usecase-reqs-summary], this model can also be treated as an application of the I2RS network topology model [RFC8345] in the scenario of Data center network management. It can also act as a service topology when mapping network elements at the fabric layer to elements of other topologies, such as L3 topologies as defined in [RFC8346].

By using the fabric topology model defined in this document, people can treat a fabric as a holistic entity and focus on characteristics of a fabric (such as encapsulation type, gateway type, etc.) as well as its connections to other fabrics while putting the underlay topology aside. As such, clients can consume the topology information at the fabric level with no need to be aware of the entire set of links and nodes in the corresponding underlay networks. A fabric topology can be configured by a network administrator using the controller by adding physical devices and links into a fabric. Alternatively, fabric topology can be learned 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]. 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.

2.1. Terminology

Fabric: also known as a POD, is a module of network, compute, storage, and application components that work together to deliver networking services. It represents a repeatable design pattern. Its

components maximize the modularity, scalability, and manageability of data centers.

3. Model Overview

This section provides an overview of the data center 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.

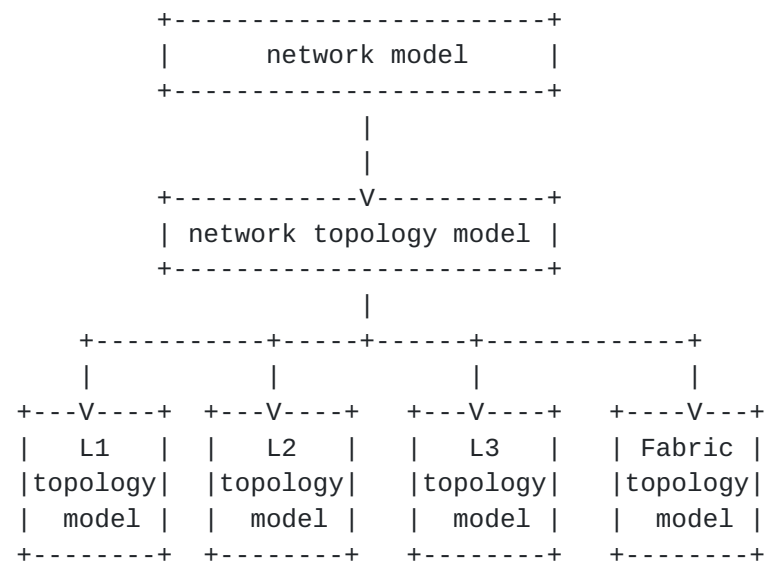

Figure 1: The network data model structure

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. It provides a fabric-based topology view for data center applications.

3.2.1. Fabric Topology

In the fabric topology module, a fabric is modeled as a node of a network, as such the fabric-based data center network consists of a set of fabric nodes and their connections. The following depicts a snippet of the definitions to show the main structure of the model. The notation syntax follows [RFC8340].

```
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:

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

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

- Termination points (in network topology module) are augmented with fabric port attributes defined in a container. The "termination-point" here is used to represent a fabric "port" that provides connections to other nodes, such as an internal device, another fabric externally, or end hosts.

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

3.2.2. Fabric node extension

As an atomic network, a fabric itself is composed of a set of network elements i.e. devices, and related links. The configuration of a fabric is contained under the "fabric-attributes" container depicted as follows. The notation syntax follows [RFC8340].

```
              +--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
```

In the module, additional data objects for fabric 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 the gateway-mode type (centralized or distributed) and traffic-behavior (whether an Access Control Lists (ACLs) is needed for the traffic). Also, it includes a list of device-nodes and related links as supporting-nodes to form a fabric network. These device nodes and links are represented as leaf-refs of existing nodes and links in the underlay topology. For the device-node, the "role" object is defined to represent the role of a device within the fabric, such as "SPINE" or "LEAF", which should work together with the gateway-mode.

3.2.3. Fabric termination-point extension

Since a fabric can be considered as a node, "termination-points" can represent fabric "ports" that connect to other fabrics, end hosts, as well as devices inside the fabric.

As such, the set of "termination-points" of a fabric indicate all connections of the fabric, including its internal connections, interconnections with other fabrics, and connections to end hosts.

The structure of fabric ports is as follows. The notation syntax follows [RFC8340].

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 including fabric 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 a port.

The termination point information is learned from the underlay networks, not configured by the fabric topology layer.

4. Fabric YANG Module

This module imports typedefs from [RFC8345], and it references [RFC7348] and [RFC8344].

```
<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) 2018 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
        (https://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@2018-02-11.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
        "RFC 8345:A Data Model for Network Topologies";
    }

    import ietf-network-topology {
        prefix nt;

        reference
        "RFC 8345:A Data Model for Network Topologies";
    }

    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 date 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) 2018 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
        (https://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 "2018-02-11"{
        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-05";
    }

    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, no policy is enforced.";
                }
```

```
                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-list 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 vni(VXLAN Network Identifier
            defined in RFC 7348)s 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 fabric
topology.";
```

```
        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. The subtrees and data nodes and their sensitivity/vulnerability in the ietf-dc-fabric-topology module are as follows:

fabric-attributes: A malicious client could attempt to sabotage the configuration of important fabric attributes, such as device-nodes or 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. The subtrees and data nodes and their sensitivity/vulnerability in the ietf-dc-fabric-topology module are as follows:

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, Donald E. Eastlake, Xufeng Liu, Susan Hares, Wei Song, Luis M. Contreras and Benoit Claise.

8. References

8.1. Normative References

[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.

[RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", August 2014, <http://www.rfc-editor.org/info/rfc7348>.

[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>.

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture", RFC 8342, March 2018.

[RFC8344] Bjorklund, M., "A YANG Data Model for IP Management", RFC 8344, March 2018, <http://www.rfc-editor.org/info/rfc8344>.

[RFC8345] Clemm, A., Medved, J., Tkacik, T., Varga, R., Bahadur, N., and H. Ananthakrishnan, "A YANG Data Model for Network Topologies", RFC 8345, March 2018, <http://www.rfc-editor.org/info/rfc8345>.

[RFC8346] 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", RFC 8346, March 2018, <http://www.rfc-editor.org/info/rfc8346>.

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, May 2015.

[I-D.draft-ietf-nvo3-geneve] Gross, J., Ganga, I., and T. Sridhar, "Geneve: Generic Network Virtualization Encapsulation", I-D draft-ietf-nvo3-geneve-06, March 2018.

[RFC8340] Bjorklund, M. and L. Berger, "YANG Tree Diagrams", RFC 8340, March 2018, <http://www.rfc-editor.org/info/rfc8340>.

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 [RFC8342]. 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@2018-02-11.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
    "RFC 8345:A Data Model for Network Topologies";
}
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 date 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 date 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) 2018 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(https://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 "2018-02-11" {
        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-05";
    }

    //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-list 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 are included within the fabric.";
        uses link-attributes;
      }
```

```
      list device-ports {
        key port-ref;
        description "Ports that are included within 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