Stream: Internet Engineering Task Force (IETF)
RFC: 8542
Category: Standards Track
Published: March 2019
ISSN: 2070-1721

Authors: Y. Zhuang *Huawei*, D. Shi *Huawei*, R. Gu *China Mobile*, H. Ananthakrishnan *Netflix*

A YANG Data Model for Fabric Topology in Data-Center Networks

Abstract

This document defines a YANG data model for fabric topology in data-center networks and represents one possible view of the data-center fabric. This document focuses on the data model only and does not endorse any kind of network design that could be based on the abovementioned model.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8542.

Copyright Notice

Table of Contents

1. Introduction

A data-center (DC) network can be composed of single or multiple fabrics, which are also known as Points Of Delivery (PODs). These fabrics may be heterogeneous due to implementation of different technologies when a DC network is upgraded or new techniques and features are rolled out. For example, within a DC network, Fabric A may use Virtual eXtensible Local Area Network (VXLAN) while Fabric B may use VLAN. 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 the NVO3 Working Group, such as Geneve [GENEVE]. The configuration and management of such DC networks with heterogeneous fabrics could result in considerable complexity.

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. The advantage of this method is to make the overall management tasks flexible and easy to extend in the future.

As a basis for DC fabric management, this document defines a YANG data model [RFC6020] [RFC7950] for a possible view of the 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], this model can also be treated as an application of the Interface to the Routing System (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 its characteristics (such as encapsulation type and gateway type) 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 and Acronyms

2.1. Key Words

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2.2. Terminology

POD: 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.

Fabric: composed of several PODs to form a data-center network.

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 Figure 1.

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

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 and VXLAN. 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-dc-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-dc-fabric-topology", is defined 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 the 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 (that is, a set of nodes and links that composes a POD and also supports a single overlay/underlay instance), 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?      fabrictypes: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, and 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 List (ACL) is needed for the traffic). Also, it includes a list of device nodes and related links as "supporting-node" 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 and end hosts, as well as devices inside the fabric.

As such, the set of "termination-points" of a fabric indicate all of its connections, 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].

```
  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)?
```

This structure augments the termination points (in the 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), and port type (L2 interface, L3 interface). By defining the device port as a tp-ref, a fabric port can be mapped to a device node in the underlay network.

Additionally, 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 Modules

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

```
<CODE BEGINS> file "ietf-dc-fabric-types@2019-02-25.yang"

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

  import ietf-network {
    prefix nw;
    reference
      "RFC 8345: A YANG Data Model for Network Topologies";
  }

  organization
    "IETF I2RS (Interface to the Routing System) Working Group";
  contact
    "WG Web:    <https://datatracker.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) 2019 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 RFC 8542;
     see the RFC itself for full legal notices.";

  revision 2019-02-25 {
    description
      "Initial revision.";
    reference
      "RFC 8542: A YANG Data Model for Fabric Topology
       in Data-Center Networks";
  }

  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
      "Ethernet";
  }

  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
      "10Mbps";
  }

  identity bw-100M {
    base bandwidth;
    description
      "100Mbps";
  }

  identity bw-1G {
    base bandwidth;
    description
      "1Gbps";
  }

  identity bw-10G {
    base bandwidth;
    description
      "10Gbps";
  }

  identity bw-25G {
    base bandwidth;
    description
      "25Gbps";
  }

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

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

  identity bw-400G {
    base bandwidth;
    description
      "400Gbps";
  }

  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.";
  }

  identity service-capability {
    description
      "Base for the service of the fabric ";
  }

  identity ip-mapping {
    base service-capability;
    description
      "NAT.";
  }

  identity acl-redirect {
    base service-capability;
    description
      "ACL redirect, which can provide a Service Function Chain
(SFC).";
  }

  identity dynamic-route-exchange {
    base service-capability;
    description
      "Dynamic route exchange.";
  }

  /*
   * Typedefs
   */

  typedef fabric-id {
    type nw:node-id;
    description
      "An identifier for a fabric in a topology.
       This identifier can be generated when composing a fabric.
       The composition of a fabric can be achieved by defining an
       RPC, which is left for vendor specific implementation
       and not provided in this model.";
  }

  typedef service-capabilities {
    type identityref {
      base service-capability;
    }
    description
      "Service capability of the fabric";
  }

  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@2019-02-25.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 YANG Data Model for Network Topologies";
  }
  import ietf-network-topology {
    prefix nt;
    reference
      "RFC 8345: A YANG Data Model for Network Topologies";
  }
  import ietf-dc-fabric-types {
    prefix fabrictypes;
    reference
      "RFC 8542: A YANG Data Model for Fabric Topology in
       Data-Center Networks";
  }

  organization
    "IETF I2RS (Interface to the Routing System) Working Group";
  contact
    "WG Web:    <https://datatracker.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) 2019 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 RFC 8542;
     see the RFC itself for full legal notices.";

  revision 2019-02-25 {
    description
      "Initial revision.";
    reference
      "RFC 8542: A YANG Data Model for Fabric Topology
       in Data-Center Networks";
  }

  //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 centralized
             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 means no policy is needed
             for all traffic";
        }
        enum policy-driven {
          description
            "Policy driven means policy is
             needed for the traffic; otherwise, the traffic
             will be discarded.";
        }
      }
      default "normal";
      description
        "Traffic behavior of the fabric";
    }
    leaf-list capability-supported {
      type fabrictypes:service-capabilities;
      description
        "It provides a list of supported services of the
         fabric.  The service-capabilities is defined as
         identity-ref.  Users can define more services
         by defining new identities.";
    }
  }

  grouping device-attributes {
    description
      "device attributes";
    leaf device-ref {
      type fabrictypes:node-ref;
      description
        "The device that the fabric includes that refers
         to a node in another topology.";
    }
    leaf-list role {
      type fabrictypes:device-role;
      default "fabrictypes:leaf";
      description
        "It is a list of device roles to represent the roles
         that a device plays within a POD, such as SPINE,
         LEAF, Border, or Border-Leaf.
         The device role is defined as identity-ref.  If more
         than 2 stages are used for a POD, users can
         define new identities for the device role.";
    }
  }

  grouping link-attributes {
    description
      "Link attributes";
    leaf link-ref {
      type fabrictypes:link-ref;
      description
        "The link that the fabric includes that refers to
         a link in another topology.";
    }
  }
  grouping port-attributes {
    description
      "Port attributes";
    leaf port-ref {
      type fabrictypes:tp-ref;
      description
        "The port that the fabric includes that refers to
         a termination-point in another topology.";
    }
    leaf port-type {
      type fabrictypes:port-type;
      description
        "Port type is defined as identity-ref.  The current
         types include ethernet or serial.  If more types
         are needed, developers can define new identities.";
    }
    leaf bandwidth {
      type fabrictypes:bandwidth;
      description
        "Bandwidth of the port.  It is defined as identity-ref.
         If more speeds are introduced, developers can define
         new identities for them.  Current speeds include 1M, 10M,
         100M, 1G, 10G, 25G, 40G, 100G, and 400G.";
    }
  }

  grouping fabric-attributes {
    description
      "Attributes of a fabric";
    leaf fabric-id {
      type fabrictypes:fabric-id;
      description
        "An identifier for a fabric in a topology.
         This identifier can be generated when composing a fabric.
         The composition of a fabric can be achieved by defining an
         RPC, which is left for vendor-specific implementation and
         not provided in this model.";
    }
```

```
      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
          "The range of the VXLAN Network Identifier
           (VNI) defined in RFC 7348 that the POD uses.";
        leaf min {
          type int32;
          description
            "The lower-limit VNI.";
        }
        leaf max {
          type int32;
          description
            "The upper-limit VNI.";
        }
      }
      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 are included in a fabric.";
        uses device-attributes;
      }
      list device-links {
        key "link-ref";
        description
          "Links that are included within a fabric.";
        uses link-attributes;
      }
      list device-ports {
        key "port-ref";
        description
          "Ports that are included in the fabric.";
        uses port-attributes;
      }
    }

    // augment statements

    augment "/nw:networks/nw:network/nw:network-types" {
      description
        "Introduce a new network type for fabric-based 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]:

Name: ietf-dc-fabric-types

Namespace: urn:ietf:params:xml:ns:yang:ietf-dc-fabric-types

Prefix: fabrictypes

Reference: RFC 8542

Name: ietf-dc-fabric-topology

Namespace: urn:ietf:params:xml:ns:yang:ietf-dc-fabric-topology

Prefix: fabric

Reference: RFC 8542

Name: ietf-dc-fabric-topology-state

Namespace: urn:ietf:params:xml:ns:yang:ietf-dc-fabric-topology-state

Prefix: sfabric

Reference: RFC 8542

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 [RFC8446].

The Network Configuration Access Control Model (NACM) [RFC8341] 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.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. 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 and name.

7. References

7.1. 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 , <https://www.rfc-editor.org/info/rfc2119>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004 , <https://www.rfc-editor.org/info/rfc3688>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010 , <https://www.rfc-editor.org/info/rfc6020>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011 , <https://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011 , <https://www.rfc-editor.org/info/rfc6242>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016 , <https://www.rfc-editor.org/info/rfc7950>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017 , <https://www.rfc-editor.org/info/rfc8040>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017 , <https://www.rfc-editor.org/info/rfc8174>.

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018 , <https://www.rfc-editor.org/info/rfc8341>.

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018 , <https://www.rfc-editor.org/info/rfc8342>.

[RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March 2018 , <https://www.rfc-editor.org/info/rfc8345>.

[RFC8346] Clemm, A., Medved, J., Varga, R., Liu, X., Ananthakrishnan, H., and N. Bahadur, "A YANG Data Model for Layer 3 Topologies", RFC 8346, DOI 10.17487/RFC8346, March 2018 , <https://www.rfc-editor.org/info/rfc8346>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018 , <https://www.rfc-editor.org/info/rfc8446>.

7.2. Informative References

[GENEVE] Gross, J., Ganga, I., and T. Sridhar, "Geneve: Generic Network Virtualization Encapsulation", Work in Progress, draft-ietf-nvo3-geneve-12, 11 March 2019.

[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", RFC 7348, DOI 10.17487/RFC7348, August 2014 , <https://www.rfc-editor.org/info/rfc7348>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018 , <https://www.rfc-editor.org/info/rfc8340>.

[RFC8344] Bjorklund, M., "A YANG Data Model for IP Management", RFC 8344, DOI 10.17487/RFC8344, March 2018 , <https://www.rfc-editor.org/info/rfc8344>.

Appendix A. Non-NMDA-State Modules

The YANG module, ietf-dc-fabric-topology, 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 cases 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 document in conjunction with non-NMDA-compliant implementations, a corresponding companion module needs to be introduced as well. This companion module, ietf-dc-fabric-topology-state, mirrors ietf-dc-fabric-topology. However, the ietf-dc-fabric-topology-state 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-dc-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. 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@2019-02-25.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
      "RFC 8542: A YANG Data Model for Fabric Topology in
       Data-Center Networks";
  }

  organization
    "IETF I2RS (Interface to the Routing System) Working Group";
  contact
    "WG Web:    <https://datatracker.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 either is learned
     or results from applying topology 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 implementation infrastructure supports
     the Network Management Datastore Architecture (NMDA).

     Copyright (c) 2019 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 RFC 8542;
     see the RFC itself for full legal notices.";

  revision 2019-02-25 {
    description
      "Initial revision.";
    reference
      "RFC 8542: A YANG Data Model for Fabric Topology in
       Data-Center Networks";
  }

  //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 centralized
             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 means no policy is needed
             for all traffic";
        }
        enum policy-driven {
          description
            "Policy driven means policy is
             needed for the traffic; otherwise, the traffic
             will be discarded.";
        }
      }
      default "normal";
      description
        "Traffic behavior of the fabric";
    }
    leaf-list capability-supported {
      type fabrictypes:service-capabilities;
      description
        "It provides a list of supported services of the
         fabric.  The service-capabilities is defined as
         identity-ref.  Users can define more services
         by defining new identities.";
    }
  }

  grouping device-attributes {
    description
      "device attributes";
    leaf device-ref {
      type fabrictypes:node-ref;
      description
        "The device that the fabric includes that refers
         to a node in another topology.";
    }
    leaf-list role {
      type fabrictypes:device-role;
      default "fabrictypes:leaf";
      description
        "It is a list of device roles to represent the roles
         that a device plays within a POD, such as SPINE,
         LEAF, Border, or Border-Leaf.
         The device role is defined as identity-ref.  If more
         than 2 stages are used for a POD, users can
         define new identities for the device role.";
    }
  }

  grouping link-attributes {
    description
      "Link attributes";
    leaf link-ref {
      type fabrictypes:link-ref;
      description
```

```
        "The link that the fabric includes that refers to
         a link in another topology.";
    }
  }

  grouping port-attributes {
    description
      "Port attributes";
    leaf port-ref {
      type fabrictypes:tp-ref;
      description
        "The port that the fabric includes that refers to
         a termination-point in another topology.";
    }
    leaf port-type {
      type fabrictypes:port-type;
      description
        "Port type is defined as identity-ref.  The current
         types include ethernet or serial.  If more types
         are needed, developers can define new identities.";
    }
    leaf bandwidth {
      type fabrictypes:bandwidth;
      description
        "Bandwidth of the port.  It is defined as
         identity-ref.  If more speeds are introduced,
         developers can define new identities for them.  Current
         speeds include 1M, 10M, 100M, 1G, 10G,
         25G, 40G, 100G, and 400G.";
    }
  }

  grouping fabric-attributes {
    description
      "Attributes of a fabric";
    leaf fabric-id {
      type fabrictypes: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
        "The range of the VXLAN Network
         Identifier (VNI) defined in RFC 7348 that the POD uses.";
      leaf min {
        type int32;
        description
          "The lower-limit VNI.";
      }
      leaf max {
        type int32;
        description
          "The upper-limit VNI.";
      }
    }
    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 are included in a fabric.";
      uses device-attributes;
    }
    list device-links {
      key "link-ref";
      description
        "Links that are included within a fabric.";
      uses link-attributes;
    }
    list device-ports {
      key "port-ref";
      description
        "Ports that are included in the fabric.";
      uses port-attributes;
    }
  }

  // augment statements

  augment "/nws:networks/nws:network/nws:network-types" {
    description
      "Introduce a 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>
```

Acknowledgements

We wish to acknowledge the helpful contributions, comments, and suggestions that were received from Alexander Clemm, Donald E. Eastlake 3rd, Xufeng Liu, Susan Hares, Wei Song, Luis M. Contreras, and Benoit Claise.

Authors' Addresses

China
210012
JiangsuNanjing
101 Software Avenue, Yuhua District
Huawei
Yan Zhuang
Email: zhuangyan.zhuang@huawei.com

China
210012
JiangsuNanjing
101 Software Avenue, Yuhua District
Huawei
Danian Shi
Email: shidanian@huawei.com

China
100053
BeijingBeijing
32 Xuanwumen West Ave, Xicheng District
China Mobile
Rong Gu
Email: gurong_cmcc@outlook.com

Hariharan Ananthakrishnan
Netflix
Email: hari@netflix.com