I2RS Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 3, 2016

Y. Zhuang
D. Shi
Huawei

June 1, 2016

A YANG Data Model for Fabric Topology in Data Center Network
draft-zhuang-i2rs-yang-dc-fabric-network-topology-00

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 http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 3, 2016.

Copyright Notice

Table of Contents

1. Introduction

Currently, a data center network is composed of single or multiple fabrics which are also known as PODs (a Point Of Delivery), which may be heterogeneous due to implementation of different technologies within these fabrics. For example, within a DC network, Fabric A may use VXLAN while Fabric B may use VLAN. The management of the entire heterogeneous network will be sophisticated and complex especially when the DC network scales.

However, in this heterogeneous DC network, fabric can be considered to be an atomic structure to provide network management and services, as well as capacity scaling. From this point, the management of the entire DC network can be broken down into management of these fabrics as to provide easier management and flexibility and scalability.

With this purpose, this document defines a YANG data model for the Fabric topology of Data Center Networks by using YANG [6020][I-D.draft-ietf-netmod-rfc6020bis-09]. 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 a DC fabric.

This model defines the generic configuration and operational state for the DC fabric common to all vendor implementation, which can be extended by vendors with specific information. This model can be

used by the network elements to expose to a network controller and capture their understanding of the overall network topology. Also, it can be used by a network controller to represent its view of the fabric topology that it controls and expose it to applications for network management for Data-center networks.

Moreover, in 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 act as a service topology in conjunction with the existing I2RS topologies. As a service topology, this DC fabric topology can be further connected with network elements in the underlay topologies, such as L3 topology defined in [I.D. draft-ietf-i2rs-yang-l3-topology-01]. By using the DC fabric topology, people can describe characteristics of a DC fabric (such as encapsulation, spine/leaf node, and gateway type, etc.). Also, clients can consume the topology information at fabric level, while no need to parse entire set of links and nodes in underlay networks.

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
```

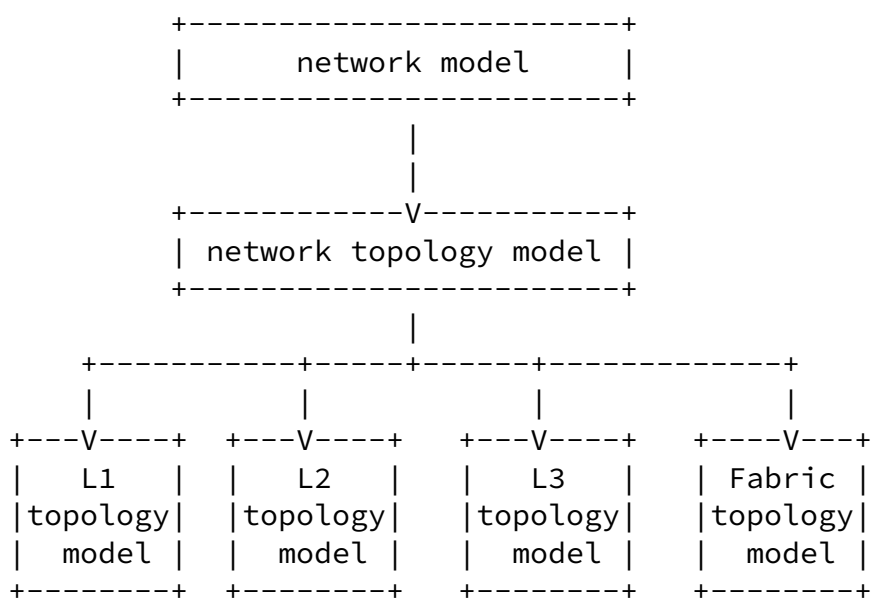
In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

3. Model Overview

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

3.1. 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).



From the perspective of resource management and service provisioning, 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

The fabric topology model is designed to be generic and can be applied to data center fabrics built with different technologies, such as VLAN, VXLAN etc al.

The structure of "ietf-fabric-topology" data model is depicted in the following diagram.


```
augment /nw:networks/nw:network/nw:network-types:
   +--rw fabric-network!
augment /nw:networks/nw:network:
   +--rw fabric-network-attributes
      +--rw name?           string
      +--rw fabric-id?      fabric-id
      +--rw type?           fabrictype:underlayer-network-type
      +--rw description?    string
      +--rw options
         +--rw gateway-mode?       enumeration
         +--rw traffic-behavior?   enumeration
augment /nw:networks/nw:network/nw:node:
   +--rw node-ref?    fabrictype:node-ref
   +--rw role?        fabrictype:device-role
augment /nw:networks/nw:network/nt:link:
```

```
   +--rw link-ref?   fabrictype:link-ref
augment /nw:networks/nw:network/nw:node/nt:termination-point:
   +--rw lport-attributes
      +--rw lport-uuid?         yang:uuid
      +--rw name?               string
      +--rw role?               port-role
      +--rw layer-1-info
      |  +--rw location?   tp-ref
      +--rw layer-2-info
      |  +--rw access-type?      access-type
      |  +--rw access-segment?   uint32
      +--rw layer-3-info
      |  +--rw ip?               inet:ip-address
      |  +--rw network?          inet:ip-prefix
      |  +--rw mac?              yang:phys-address
      |  +--rw forward-enable?   boolean
      +--rw fabric-acl* [fabric-acl-name]
      |  +--rw fabric-acl-name    string
      +--rw underlayer-ports* [port-ref]
         +--rw tp-ref?        -> /nd:networks/network[nd:network-id=current()/.
                 network-ref]/node[nd:node-id=current()/../node-ref]/lnk:termin
         +--rw node-ref?      -> /nd:networks/network[nd:network-id=current()/
                 ../network-ref]/node/node-id
         +--rw network-ref?   -> /nd:networks/network/network-id
```

The fabric-topology 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.

- Additional topology attributes are introduced, defined in a grouping, which augments the "network" list of the network module.

The attributes include a Fabric name, fabric-id, type of the underlay network, descriptions of the fabric as well as a set of options defined in a container. The options container includes leaves of the gateway-mode (centralized or distributed) and traffic-behavior (whether acl needed for the traffic).

- Additional data objects for nodes are introduced by augmenting the "node" list of the network module. New objects include node id of

a node in the topology and its role, such as "SPINE" node or "LEAF" node, which should work together with gateway-mode.

- Termination points (in network topology module) are augmented with logical port attributes defined in a container. It contains common information of name, id, role of the port (such as port connected to endpoint, or port connected to intra-fabric communication, etc al). Besides, it also carries layering information for a port with containers of layer 1, 2 and 3 information. A set of acl for the fabric is also defined. At last, it provides a set of underlay ports that the logical fabric port can be mapped to.

In addition, we also define several rpc operations for fabric managements within data center networks to get and configure fabrics and obtain a response. The YANG module defines 7 rpc commands for this management, including getting all fabric information, composing a fabric, decomposing a fabric, adding a node to a fabric, removing a node from a fabric, adding a link to a fabric and removing a link from a fabric.

4. Fabric YANG Module

```
<CODE BEGINS> file "ietf-fabric-types@2016-02-26.yang"
module ietf-fabric-types {

yang-version 1;
namespace "urn:ietf:params:xml:ns:yang:ietf-fabric-types";
prefix ftypes;

import ietf-inet-types {
            prefix "inet";
    }
import ietf-yang-types {
            prefix "yang";
            revision-date "2013-07-15";
    }
    import ietf-network-topology {
            prefix nt;
    }
```



```
    organization
```

```
"Huawei Technologies";
    contact
"shidanian@huawei.com";
description
    "This module contains a collection of YANG definitions for Fabric netwo

revision "2016-02-26" {
    description
        "Initial revision of fabric network service.";
            reference
                    "";
}
    // identity statements
identity fabric-type {
            description "base fabric type";
}
identity vxlan-fabric {
    base fabric-type;
            description "vxlan-fabric fabric";
}
    // typedef statements.
    // ref nodes are all changed to refer to network/network-topology data
typedef node-ref {
    type instance-identifier;
            description "A reference to a node in topology";
}
typedef tp-ref {
    type instance-identifier;
            /*type leafref {
        path "/network-topology/topology/node/termination-point/tp-id";
    }*/
    description "A reference to a termination point in topology";
}
typedef link-ref {
    type instance-identifier;
            description "A reference to a link in topology";
}
typedef device-role {

    type enumeration {
        enum SPINE {
            description "";
        }
        enum LEAF {
            description "";
        }
    }
```

```
                description "device role type";
    }
    typedef port-role {

        type enumeration {
            enum NNI {
                description "the port used for fabric intra communiation.";
            }
            enum UNI {
                description "the port used for Endpoint to access fabric.";
            }
        }
                description "port role type";
    }
    typedef underlayer-network-type {

        type enumeration {
            enum VXLAN {
                description "vxlan";
            }
            enum TRILL {
                description "trill";
            }
        }
                description "";
    }
    typedef layer2-protocol-type-enum {
        type enumeration {
            enum VLAN{
                                description "vlan";
                        }
            enum VXLAN{
                                description "vxlan";
                        }
            enum TRILL{
                                description "trill";
                        }
            enum NvGRE{
                                description "nvgre";
                        }
        }
                description "l2 protocol type";
    }
    typedef access-type {
        type enumeration {
            enum exclusive{
```

```
                                description "exclusive";
                        }
```



```
            enum vlan{
                                description "vlan";
                        }
        }
                description "access type";
    }
        // grouping statements
    grouping acl-list {

        list fabric-acl {
            key fabric-acl-name;
            leaf fabric-acl-name {
                type string;
                                description "fabric acl name";
            }
                        description "list of fabric acl";
        }
                description "fabric acl list";
    }

    grouping logic-switch {
        description "logic switch attributes";
        leaf lsw-uuid {
            type yang:uuid;
                        description "logic switch id";
        }
        leaf name {
            type string;
                        description "name of the logic switch";
        }
        leaf segment-id {
            type uint32;
                        description "segement id";
        }
        leaf network {
            type inet:ip-prefix;
                        description "ip prefix";
        }
        uses acl-list;
```

```
    }
    grouping logic-router {
        description "logic router attributes";
        leaf lr-uuid {
            type yang:uuid;
                        description "logic router id";
        }
        leaf name {
            type string;
```



```
                        description "logic router name";
        }
        leaf vrf-ctx {
            type uint32;
                        description "vrf-context if vxlan";
        }
        uses acl-list;
    }
    grouping logic-port {
        description "logic port attributes";
        leaf lport-uuid {
            type yang:uuid;
                        description "logic port id";
        }
        leaf name {
            type string;
                        description "logic port name";
        }
        leaf role {
            type port-role;
                        description "role";
        }
        container layer-1-info {
                        description "layer 1 infor";
            leaf location {
                type tp-ref;
                                description "tp ref to layer 1 topo";
            }
        }
        container layer-2-info {
                        description "layer 2 infor";
            leaf access-type {
```

```
                type access-type;
                                description "access type";
            }
            leaf access-segment {
                type uint32;
                                description "access segment";
            }
        }
        container layer-3-info {
                        description "layer 3 info";

            leaf ip {
                type inet:ip-address;
                                description "ip address";
            }
            leaf network {
```



```
                type inet:ip-prefix;
                                description "ip prefix";
            }
            leaf mac {
                type yang:phys-address;
                                description "mac address";
            }
            leaf forward-enable {
                type boolean;
                                description "";
            }
        }
        uses acl-list;

                list underlayer-ports {
                        key port-ref;
                        uses nt:tp-ref;
                        description "underlayer-network-type ports infor";
                }
    }
}
    <CODE ENDS>

        <CODE BEGINS> file "ietf-fabric-topology@2016-04-13.yang"
        module ietf-fabric-topology {
```

```
    yang-version 1;
    namespace "urn:ietf:params:xml:ns:yang:ietf-fabric-topology";
    prefix fabric;

    import ietf-network {
                prefix nw;
        }
        import ietf-network-topology {
                prefix nt;
        }
    import ietf-fabric-types {
                        prefix fabrictype;
                        revision-date "2016-02-26";
        }
        organization
    "Huawei Technologies";
    contact
    "shidanian@huawei.com";
    description
        "This module contains a collection of YANG definitions for Fabric.";

        revision "2016-04-13"{
```



```
                description
                        "intial version";
                reference
                        "";
        }

   // typedef statements
    typedef fabric-id {
        type nw:node-id;
        description
            "An identifier for a fabric in a topology.
            The identifier is generated by create-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-network-attributes {
    description "fabric Topology scope attributes";
    container fabric-network-attributes {
      description "Containing fabric network attributes";
      leaf name {
          type string;
                      description "name of the fabric";
      }

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

      leaf type {
          type fabrictype:underlayer-network-type;
          description "The type of physcial network that implements this fabr
                      Examples are vlan, and trill.";
      }

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



```
      container options {
                      description "options for the fabric";
          uses fabric-options;
      }
    }
  }

  grouping fabric-options {
      description "options for fabric";
      leaf gateway-mode {
          type enumeration {
```

```
                enum centralized {
                    description "";
                }
                enum distributed {
                    description "";
                }
            }
            default "distributed";

                        description "gateway mode";
        }

                leaf traffic-behavior {

                        type enumeration {
                                enum normal {
                                        description "";
                                }
                                enum need-acl {
                                        description "";
                                }
                        }
                        default "normal";
                        description "traffic behavior";
                }
    }

    grouping fabric-device-attributes {
        description "fabric node attributes";
        leaf node-ref {
            type fabrictype:node-ref;
                        description "";
        }
                //uses nw:node-ref;
        leaf role {
            type fabrictype:device-role;
                        description "role of the device, such as leaf or spine"
```



```
        }
    }

    grouping fabric-link-attributes {
```

```
        description "fabric link attribute";
        leaf link-ref {
            type fabrictype:link-ref;
                        description "";
        }
    }

        // 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" {
                when "nw:network-types/fabric-network-type" {
                        description
                        "Augmentation parameters apply only for networks with
                        fabric network topo";
                }
                description
        "Configuration parameters for fabric network";
                uses fabric-network-attributes;
   }
    augment "/nw:networks/nw:network/nw:node" {
                when "/nw:networks/nw:network/nw:network-types/fabric-network-t
                        description "Augmentation parameters apply only for net
       fabric topology";
                }
                description "Configuraion parameters for nodes";
            uses fabric-device-attributes;
    }

        augment "/nw:networks/nw:network/nt:link" {
      when "/nw:networks/nw:network/nw:network-types/fabric-network-type" {
        description
          "Augmentation parameters apply only for networks
           with fabric topology";
      }
      description "Augment fabric topology link information";
      uses fabric-link-attributes;
    }

    augment "/nw:networks/nw:network/nw:node/nt:termination-point" {
        when "/nw:networks/nw:network/nw:network-types/fabric-network-type"{
```

```
                        description
                        "Augmentation parameters apply only for networks
           with fabric topology";
                }
                description
        "Augment fabric topology termination point configuration";
        container lport-attributes {
                        description "fabric tp configuraiton";
            uses fabrictype:logic-port;
        }
    }

    /*******************************RPC*************************************
    rpc get-all-fabric {
        description "get all fabrics";
        output {
            leaf-list fabric-id {
               type fabric-id;
                            description "fabric id";
            }
        }
    }

    rpc compose-fabric {
        description "compose a fabric";
        input {
            uses fabric-network-attributes;
        }
        output {
            leaf fabric-id {
                type fabric-id;
                                description "fabric id";
            }
        }
    }

    rpc decompose-fabric {
        description "decompose a fabric";
        input {
            leaf fabric-id {
                type fabric-id;
                                description "fabric id";
            }
        }
    }

    rpc add-node-to-fabric {
```

```
        description "add node to a fabric";
```



```
        input {
            leaf fabric-id {
                type fabric-id;
                                description "fabric id";
            }
            uses fabric-device-attributes;
        }
    }

    rpc rm-node-from-fabric {
        description "remove a node from a fabric";
        input {
            leaf fabric-id {
                type fabric-id;
                                description "fabric id";
            }
                        uses nw:node-ref;
        }
    }

    rpc add-link-to-fabric {
        description "add a link to a fabric";
        input {
            leaf fabric-id {
                type fabric-id;
                                description "fabric id";
            }
            uses fabric-link-attributes;
        }
    }

    rpc rm-link-from-fabric {
        description "remove a link from a fabric";
        input {
            leaf fabric-id {
                type fabric-id;
                                description "fabric id";
            }
                        uses nt:link-ref;
        }
```

```
    }
}
        <CODE ENDS>
```


5. Security Consideration

TBD

6. Acknowledgements

7. References

7.1. Informative 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-network-topo-02, December 2015.

[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-02, December 2015.

[I-D.draft-ietf-netmod-rfc6020bis-09]
Bjorklund, M., "The YANG 1.1 Data Modeling Language", I-D draft-ietf-netmod-rfc6020bis-09, December 2015.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020,

October 2010.

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

7.2. 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.

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