Network Working Group
Internet Draft
Intended status: Standard Track
Expires: November 2015

D. Zhang

A. Zaalouk
EICT
Y. Cheng, Ed
China Unicom
J. Lindblad
Tail-F
M. Klyus
NetCracker

May 11, 2015

VPN Service Management YANG Data Model
draft-zaalouk-supa-vpn-service-management-model-05

Abstract

Currently new services create new opportunities for both network providers and service providers. Simplified Use of Policy Abstractions (SUPA) was proposed to develop a model that abstracts network resources and services and a methodology by which the management and monitoring of network services can be done using standardized policy rules. This document defines a VPN service management yang data model and gives an example for DDC use case.

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), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

This Internet-Draft will expire on Expires November 5, 2015.

Table of Contents

1. Introduction

Currently new services bring new challenges and opportunities for both network and service providers. Meanwhile, legacy services

such as VPN [RFC4110] also need specialized management and controlling capability from the network management systems to improve the experiences for fast deployment and dynamic configuration.

Simplified Use of Policy Abstractions (SUPA) [SUPA-problem-statement] [SUPA-framework] was proposed to introduce the concepts of multi-level and multi-technology network abstractions to address the current separation between development and deployment operations. The first example that SUPA will focus on will be VPN management.

This document introduces YANG [RFC6020] [RFC6021] data models for SUPA configuration. Such models can facilitate the standardization for the interface of SUPA, as they are compatible to a variety of protocols such as NETCONF [RFC6241] and [RESTCONF]. Please note that in the context of SUPA, the term "application" refers to an operational and management applications employed, and possibly implemented, by an operator. The first configuration model is based on the first example - VPN management.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119]. 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 [RFC2119] significance.

3. Network Service Modules

In this section, several specific network service models are described based on a set of specific network services and the framework of SUPA [SUPA-framework].

3.1. Generic VPN Service Module

A virtual private network (VPN) extends a private network across a public network, such as the Internet. It enables a computer or network-enabled device to send and receive data across shared or public networks as if it were directly connected to the private

network, while benefiting from the functionality, security and management policies of the public network. [VPN]

VPN systems may be classified by multiple ways, e.g., tunnelling protocols, tunnel's termination point location, etc. A typical one among these is by the OSI layer they present to the connecting network, such as Layer 2 circuits or Layer 3 network connectivity.

In use cases of [SUPA-DDC], the links between DCs are VPNs, including L2VPN, L3VPN, etc. In this draft, before going deep into specific VPN services, a generic VPN model is firstly proposed below. It could be used by other specific VPN service models, such as L3VPN service models in [l3vpn-service-yang] and section 3.2 of this draft.

Business VPN usually refers to the order information, while Service VPN usually contains the user's CE information, these two are highly abstraction of VPN service model. From inheritance model perspective, Generic VPN is a base class of all VPN models. L2VPN, L3VPN, Service VPN, Business VPN may be inherited from this base class with corresponding extensions. The Network Manager/Controller [SUPA-framework] can provide three kinds of north interfaces: VPN business models based on the Generic VPN, L2VPN, L3VPN, respectively. This suggests that the Network Manager/Controller can directly deploy L2VPN/ L3VPN business, or VPN service without being informed about the specific VPN type. In other word, the Network Manager/Controller selects specific VPN solutions based on Generic VPN and the specific deployment environment/requirement.

```
module: ietf-supa-vpn
   +--rw vpn-instance
      +--rw vpn-instance* [instance-name]
         +--rw name                              string
         +--rw connection-type?                  enumeration
         +--rw service-type?                     enumeration
         +--rw access-management
         |  +--rw user-name        string
         |  +--rw user-password    string
         +--rw instance-oper-state?              enumeration
         +--rw instance-admin-state?             enumeration
         +--rw life-cycle-state?                 enumeration
         +--rw instance-availability-status?     enumeration
         +--rw instance-cfg-revision?            string
         +--rw instance-sla-policy?              string
         +--rw access-interface* [name]
            +--rw name                              if:interface-ref
            +--rw role                              enumeration
```

There are some candidate attributes listed below under discussion and maybe some of them will be added later:

```
         +--rw network-reliability-parameter
         +--rw performance-management-parameter
         |  +--rw pm-enable?                boolean
         +--rw fault-management-parameter
         |  +--rw alarm-enable?             boolean
         |  +--rw alarm-severity?           inet:uri
```

3.1.1. VPN YANG Model

```
<CODE BEGINS>
module ietf-supa-vpn {
  namespace "urn:ietf:params:xml:ns:yang:ietf-supa-vpn";
  // replace with IANA namespace when assigned
  prefix vpn;

  import ietf-interfaces {
    prefix "if";
  }

  organization "IETF";
  contact
    "Editor: Dacheng Zhang
    dacheng.zdc@alibaba-inc.com

    Ying Cheng
    chengying10@chinaunicom.cn
    ";

  description
    "This YANG module defines a generic VPN
    (Virtual Private Network service)
    configuration model common across all of the
    vendor implementations. ";

  revision 2015-03-24 {
    description
      "Initial revision.";
    reference " RFC 4664, RFC4364, RFC7277";
  }

  container vpn-instance {
    list vpn-instance {
      key " name";
      description
        "Indicates the name of the VPN instance created.";

      leaf name {
        type string;
        description " instance name";
      }

      leaf connection-type {
```

```
            type enumeration {
              enum L2VPN {
                value 0;
                description "L2VPN";
              }
              enum L3VPN {
                value 1;
                description "L3VPN";
              }
            }
            description
              "Indicates the type of VPN, may be L2VPN or L3VPN";
          }

          leaf service-type {
            type enumeration {
              enum full-mesh {
                value "0";
                description "full-mesh";
              }
              enum hub-spoke {
                value "1";
                description "hub-spoke";
              }
            }
            default "full-mesh";
            description "Topology type";
          }

          container access-management{
            leaf user-name {
              type string;
              mandatory true;
              description "User name for this access interface";
            }

            leaf user-password {
              type string;
              mandatory true;
              description
                "User password for the access interface. User
                name and password are listed here because VPN
                need the authentication, one typical way is to
                use user name and password, so that in CE dynamic
                migration case, CEs are able to access with
                authentication regardless location.";
            }
```

```
          description
            "This part gives a typical example for access
            management";
        }

        leaf instance-oper-state {
          type enumeration {
            enum enabled {
              value 1;
              description "VPN Service is enabled";
            }
            enum disabled {
              value 2;
              description "VPN Service is disabled";
            }
          }
          description
            "This part indicates the operational state
            of VPN, enabled or disabled.
            User may use this value to inquire the status
            of this instance";
        }

        leaf instance-admin-state {
          type enumeration {
            enum Enabled {
              value 1;
              description "VPN Service Administratively Enabled";
            }
            enum Disabled {
              value 2;
              description "VPN Service Administratively Disabled";
            }
          }
          description
            "This part indicates the administrative state
            of VPN, e.g., enabled, disabled.
            User may use this value to manage the instance";
        }

        leaf life-cycle-state {
          type enumeration {
            enum planned {
              value 1;
              description "VPN Service Planned";
            }
            enum installed {
```

```
            value 2;
            description "VPN Service Installed";
          }
          enum removed {
            value 3;
            description "VPN Service Removed";
          }
        }
        description
          "This part indicates the life cycle state
          of VPN, e.g., installed, removed, planned.";
      }

      leaf instance-availability-status {
        type enumeration {
          enum degraded {
            value 1;
            description
              "VPN Service Degraded";
          }
          enum failed {
            value 2;
            description
             "VPN Service Failed";
          }
        }
        description
          "This part indicates the availability status
          of VPN, e.g., degraded, failed.";
      }

      leaf instance-cfg-revision {
        type string;
        description
          "Indicates current service configuration revision";
      }

      leaf instance-sla-policy {
        type string;
        description
          "Indicates SLA decision policy name";
      }

      list access-interface {
        key name;
        description "Access interface name.";
```

```
          leaf name {
            type if:interface-ref;
            description "Access interface name.";
          }

          leaf role {
            when "../../service-type = 'full-mesh'" {
              description
                "The role is always center-if in a
                 hub-spoke topology, in a full-mesh
                 topology, the role needs to be
                 specified as center-if or edge-if.";
            }
            type enumeration {
              enum edge-if {
                value 0;
                description "Edge interface";
              }
              enum center-if {
                value 1;
                description "Center interface";
              }
            }
            mandatory true;
            description
              "An edge interface (edge-if) indicates... FIXME";
          }
        }
      }
      description
      "Generic VPN is a base class for VPN models.
      Other VPN models may be inherited from this base
      class with corresponding extensions.";
    }
  }
  <CODE ENDS>
```

3.2. L3VPN Service Module

A Layer 3 Virtual Private Network (L3VPN) interconnects sets of hosts and routers based on Layer 3 addresses and forwarding. L3VPN can be based on MPLS or IP technologies. L3VPN is a PE-based VPN managed by operators. L3VPN is widely used in carrier metro networks to provide VPN service for enterprise users.

A L3VPN model is a collection of L3VPN instances. A L3VPN instance contains a set of access interfaces to network devices as well as other attributes, such as routing protocol, address family, topology, and so on.

To configure a L3VPN instance, the administrator needs to specify which port(s) of a network device belongs to a L3VPN instance. Those ports and network device information can be derived from a network topology model in a network management system. The administrator also needs to specify what routing protocol needs to be configured for a L3VPN instance.

The following describes a abstracted L3VPN information model for DDC service to use, based on which users can develop applications to configure L3VPN instances for DDC service. L3SM is also working on an elaborate L3VPN service model. After it's done DDC service in SUPA may directly refer the output of L3SM. The abstracted L3VPN model here is a subset of the one of L3SM.

```
module: ietf-supa-abstracted-l3vpn
   +--rw l3vpns
      +--rw l3vpn-instance* [name]
         +--rw name                              string
         +--rw service-type?                     enumeration
         +--rw address-family?                   enumeration
         +--rw instance-oper-state?              enumeration
         +--rw instance-admin-state?             enumeration
         +--rw life-cycle-state?                 enumeration
         +--rw instance-availability-status?     enumeration
         +--rw instance-cfg-revision?            string
         +--rw instance-sla-policy?              string
         +--rw service-quality?                  enumeration
         +--rw access-interface* [name]
         |  +--rw name       if:interface-ref
         |  +--rw address    inet:ip-prefix
         |  +--rw role       enumeration
         +--rw user-name                         string
         +--rw user-password                     string
         +--rw physical-node-id                  string
         +--rw physical-access-interface         if:interface-ref
         +--rw routing
            +--rw protocol?        enumeration
            +--rw bgp-attribute
            |  +--rw remote-as-number?      inet:as-number
            |  +--rw remote-peer-address    inet:ip-address
            +--rw igp-attribute
               +--rw protocol-id?   uint32
```

3.2.1. L3VPN YANG Model

```
<CODE BEGINS>
module ietf-supa-abstracted-l3vpn {
  namespace "urn:ietf:params:xml:ns:yang:ietf-supa-abstracted-
l3vpn";
  // replace with IANA namespace when assigned
  prefix abstracted-l3vpn;

  import ietf-inet-types {
    prefix inet;
  }
  import ietf-interfaces {
    prefix if;
  }

  organization "IETF";
  contact
```

```
       "Editor: Dacheng Zhang
        dacheng.zdc@alibaba-inc.com

        Adel Zaalouk
        adel.ietf@gmail.com

        Kostas Pentikousis
        k.pentikousis@eict.de

        Jan Lindblad
        janl@tail-f.com

        Maxim Klyus
        klyus@NetCracker.com
       ";

     description
       "This YANG module an abstracted L3VPN information
        model for DDC service to useL3VPN, based on which
        users can develop applications to configure L3VPN
        instances for DDC service.

        Terms and Acronyms
          L3VPN: Layer 3 Virtual Private Network
       ";

     revision 2015-05-04 {
       description "Initial revision.";
       reference "RFC4364, RFC7277";
     }

     container l3vpns {
       description "an abstracted L3VPN information
        model for DDC service to use L3VPN.";

       list l3vpn-instance {
         key name;
         description
           "Indicates the name of the VPN instance.";

         leaf name {
           type string;
           description "L3VPN instance name.";
         }

         leaf service-type {
           type enumeration {
```

```
            enum full-mesh {
              value 0;
              description "Full-mesh topology.";
            }
            enum hub-spoke {
              value 1;
              description "Hub-spoke topology.";
            }
          }
          default "full-mesh";
          description "Topology type.";
        }

        leaf address-family {
          type enumeration {
            enum ipv4uni {
              value 0;
              description "ipv4 unicast address family.";
            }
            enum ipv6uni {
              value 1;
              description "ipv6 unicast address family.";
            }
          }
          default "ipv4uni";
          description "Address family type: IPv4 or IPv6.";
        }

        leaf instance-oper-state {
          type enumeration {
            enum enabled {
              value 1;
              description "VPN Service is enabled";
            }
            enum disabled {
              value 2;
              description "VPN Service is disabled";
            }
          }
          description
            "This part indicates the operational state
            of VPN, enabled or disabled.
            User may use this value to inquire the status
            of this instance";
        }

        leaf instance-admin-state {
```

```
            type enumeration {
              enum Enabled {
                value 1;
                description "VPN Service Administratively Enabled";
              }
              enum Disabled {
                value 2;
                description "VPN Service Administratively Disabled";
              }
            }
            description
              "This part indicates the administrative state
              of VPN, e.g., enabled, disabled.
              User may use this value to manage the instance";
          }

          leaf life-cycle-state {
            type enumeration {
              enum planned {
                value 1;
                description "VPN Service Planned";
              }
              enum installed {
                value 2;
                description "VPN Service Installed";
              }
              enum removed {
                value 3;
                description "VPN Service Removed";
              }
            }
            description
              "This part indicates the life cycle state
              of VPN, e.g., installed, removed, planned.";
          }

          leaf instance-availability-status {
            type enumeration {
              enum degraded {
                value 1;
                description
                  "VPN Service Degraded";
              }
              enum failed {
                value 2;
                description
                 "VPN Service Failed";
```

```
            }
          }
          description
            "This part indicates the availability status
            of VPN, e.g., degraded, failed.";
        }

        leaf instance-cfg-revision {
          type string;
          description
            "Indicates current service configuration revision";
        }

        leaf instance-sla-policy {
          type string;
          description
            "Indicates SLA decision policy name";
        }

        leaf service-quality {
          type enumeration {
            enum gold {
              value 0;
              description
               "Best quality of service is guaranteed.";
            }
            enum silver {
              value 1;
              description
               "Better quality of service than bronze class
               is guaranteed.";
            }
            enum bronze {
              value 2;
              description
              "Average quality of service is guaranteed.";
            }
          }
          default "bronze";
          description "Quality of service.";
        }

        list access-interface {
          key name;
          description "Access interfaces.";

          leaf name {
```

```
            type if:interface-ref;
            description "Access interface name.";
          }

          leaf address {
            type inet:ip-prefix;
            mandatory true;
            description "Access interface address, IPv4 or IPv6.";
          }

          leaf role {
            when "../../service-type = 'full-mesh'" {
              description
                "The role is always center-if in a
          hub-spoke topology, in a full-mesh
          topology, the role needs to be
          specified as center-if or edge-if.";
            }
            type enumeration {
              enum edge-if {
                value 0;
                description "Edge interface";
              }
              enum center-if {
                value 1;
                description "Center interface";
              }
            }
            mandatory true;
            description
        "center-if is only available in hub-spoke
        mode; There are two scenarios: in full
        mesh mode, the role of all the access-interface
        is edge-if, while in hub-spoke mode, the role of
        the interface of hub node is ceter-if, the one
              of spoke node is edge-if";
          }
        }

        leaf user-name {
          type string;
          mandatory true;
          description "User name for this access interface";
        }

        leaf user-password {
          type string;
```

```
          mandatory true;
          description
        "User password for the access interface in
        encrypted format. User name and password are
        listed here because VPN need the
        authentication, one typical way is to use
            user name and password, so that in CE dynamic
        migration case, CEs are able to access with
        authentication regardless location.";
        }

        leaf physical-node-id {
          type string;
          mandatory true;
          description "This is the physical node ID
          of access PE";
        }

        leaf physical-access-interface {
          type if:interface-ref;
          mandatory true;
          description " This is the Physical access
          interface of access PE.";
        }

        container routing {
          description
            "Routing configuration between PE and CE.";

          leaf protocol {
            type enumeration {
              enum bgp {
                value 0;
                description "bgp routing between PE and CE";
              }
              enum ospf {
                value 1;
                description "ospf routing between PE and CE";
              }
              enum isis {
                value 2;
                description "isis routing between PE and CE";
              }
              enum rip {
                value 3;
                description "rip routing between PE and CE";
              }
```

```
              enum static {
                value 4;
                description "static routing between PE and CE";
              }
            }
            default "ospf";
            description "Routing protocol between PE and CE.";
          }

          container bgp-attribute {
            when "../protocol = 'bgp'" {
              description "BGP specific parameters.";
            }

            leaf remote-as-number {
              type inet:as-number;
              description "Remote BGP peer as-number.";
            }

            leaf remote-peer-address {
              type inet:ip-address;
              mandatory true;
              description "Remote BGP peer address.";
            }
            description "Detailed configuration for BGP.";
          }

          container igp-attribute {
            when "../protocol = 'ospf' or
                  ../protocol = 'isis' or
                  ../protocol = 'rip'  or
                  ../protocol = 'static'" {
              description "IGP specific parameters.";
            }
            leaf protocol-id {
              type uint32;
              default 0;
              description "Valid only when protocol is IGP;
                           it can be AS number.";
            }
            description " This part defines the attribute of IGP.";
          }
        }
      }
    }
  }
  <CODE ENDS>
```

3.3. L2VPN Service Module

This section describes service model for Ethernet L2VPN.

There are different ways of classifying L2VPNs. According to the Ethernet services defined by Metro Ethernet Forum, there are mainly three types of Ethernet L2VPN service that can be provided by service providers: E-line, E-tree and E-lan.

- o E-line is point to point service.
- o E-lan is multipoint to multipoint service.
- o E-tree is multipoint to multipoint service, but the communications between some consumer sites are not allowed.

Meanwhile according to [RFC4664], there are two fundamentally different kinds of Layer 2 VPN service that a service provider could offer to a customer: Virtual Private Wire Service (VPWS) and Virtual Private LAN Service (VPLS).

- o VPWS is a L2 VPN service that provides L2 point-to-point service.
- o VPLS is a L2 VPN service that emulates LAN service across a Wide Area Network (WAN).

Based on different degrees of abstraction, the interfaces can be categorized into two kinds: a) The user can provide service type and site information to the controller and controller create VPN automatically, based on the network, maybe by pseudo wire or other method. This is the more abstracted Ethernet L2VPN service interface. b) If the provider is MPLS or MPLS-TP, controller can provide less abstracted interfaces to the user. The user can use such interfaces to control the network more agility. This document describes the former one, a more abstracted model for L2VPN.

The following describes the information model for L2VPN, based on which users can develop applications to configure L2VPN instances.

```
module: ietf-supa-l2vpn
   +--rw l2vpn-instances
      +--rw l2vpn-instance* [name]
         +--rw name                              string
         +--rw service-type                      enumeration
         +--rw instance-oper-state?              enumeration
         +--rw instance-admin-state?             enumeration
         +--rw life-cycle-state?                 enumeration
         +--rw instance-availability-status?     enumeration
         +--rw instance-cfg-revision?            string
         +--rw instance-sla-policy?              string
         +--rw service-quality?                  enumeration
         +--rw access-interface* [interface-id]
            +--rw interface-id           if:interface-ref
            +--rw role                   enumeration
            +--rw node-id                string
            +--rw interface-name         if:interface-ref
            +--rw access-type?           enumeration
            +--rw user-circuit-bitmap    string
```

3.3.1. L2VPN YANG Model

```
<CODE BEGINS>
module ietf-supa-l2vpn {
  namespace "urn:ietf:params:xml:ns:yang:ietf-supa-l2vpn";
  // replace with IANA namespace when assigned
  prefix l2vpn;

  import ietf-interfaces {
    prefix if;
  }

  organization "IETF";
  contact
    "Editor: Ying Cheng
    chengying10@chinaunicom.cn

    Vikram Choudhary
    vikschw@gmail.com
    ";

  description
    "This YANG module defines a generic service
    configuration model for L2VPN (Layer 2 Virtual
    Private Network), based on which programmers can
    develop applications to configure L2VPN instances.";
```

```
     revision 2015-04-09 {
       description
         "Initial revision";
       reference "RFC4664";
     }

     container l2vpn-instances {
       description "an abstracted L3VPN information
        model for DDC service to use L3VPN.";

       list l2vpn-instance {
         key "name";
         description
           "Indicates the name of the L2VPN instance";

         leaf name {
           type string;
           description "L2VPN instance name";
         }

         leaf service-type {
           type enumeration {
             enum e-line {
               value "0";
               description "the service type is e-line";
             }
             enum e-tree {
               value "1";
               description "the service type is e-tree";
             }
             enum e-lan {
               value "2";
               description "the service type is e-lan";
             }
           }
           mandatory true;
           description
             "The service type of this instance,
             user may choose from e-line, e-tree, e-lan";
          }

         leaf instance-oper-state {
           type enumeration {
             enum enabled {
               value 1;
               description "VPN Service is enabled";
```

```
            }
            enum disabled {
              value 2;
              description "VPN Service is disabled";
            }
          }
          description
            "This part indicates the operational state
            of VPN, enabled or disabled.
            User may use this value to inquire the status
            of this instance";
        }

        leaf instance-admin-state {
          type enumeration {
            enum Enabled {
              value 1;
              description "VPN Service Administratively Enabled";
            }
            enum Disabled {
              value 2;
              description "VPN Service Administratively Disabled";
            }
          }
          description
            "This part indicates the administrative state
            of VPN, e.g., enabled, disabled.
            User may use this value to manage the instance";
        }

        leaf life-cycle-state {
          type enumeration {
            enum planned {
              value 1;
              description "VPN Service Planned";
            }
            enum installed {
              value 2;
              description "VPN Service Installed";
            }
            enum removed {
              value 3;
              description "VPN Service Removed";
            }
          }
          description
            "This part indicates the life cycle state
```

```
            of VPN, e.g., installed, removed, planned.";
          }

          leaf instance-availability-status {
            type enumeration {
              enum degraded {
                value 1;
                description
                  "VPN Service Degraded";
              }
              enum failed {
                value 2;
                description
                 "VPN Service Failed";
              }
            }
            description
              "This part indicates the availability status
              of VPN, e.g., degraded, failed.";
          }

          leaf instance-cfg-revision {
            type string;
            description
              "Indicates current service configuration revision";
          }

          leaf instance-sla-policy {
            type string;
            description
              "Indicates SLA decision policy name";
          }

          leaf service-quality {
            type enumeration {
              enum gold {
                value 0;
                description
                 "Best quality of service is guaranteed.";
              }
              enum silver {
                value 1;
                description
                 "Better quality of service than bronze class
                 is guaranteed.";
              }
              enum bronze {
```

```
              value 2;
              description
              "Average quality of service is guaranteed.";
            }
          }
          default "bronze";
          description "Quality of service.";
        }

        list "access-interface" {
          key "interface-id";
          description "Access interface ID";

          leaf interface-id {
            type if:interface-ref;
            description "Access interface ID";
          }

          leaf role {
            when "../../instance-service-type = 'e-tree'" {
          description
          "root or leaf, only available in e-tree mode";
        }
            type enumeration {
              enum root {
                value "0";
                description "root interface";
                }
              enum leaf {
                value "1";
                description "leaf interface";
              }
            }
            mandatory true;
            description
            "the role of interface";
          }

          leaf node-id {
            type string;
            mandatory true;
            description "physical node ID of access PE";
          }

          leaf interface-name {
            type if:interface-ref;
            mandatory true;
```

```
            description
              "physical access interface name of access PE";
          }

          leaf access-type {
            type enumeration {
              enum port{
                value "0";
                description "The access type is port.";
              }
              enum vlan{
                value "1";
                description "The access type is vlan.";
              }
        }
            default "port";
            description
              "This value describes the access type,
              options are port and vlan; port is chose
              when the access is the entire port, while vlan
              means the access is the vlan of the port.";
          }

          leaf user-circuit-bitmap {
            when "../../ access-type = 'vlan'"{
          description "Only available when access-type is vlan.";
        }
            type string;
            mandatory true;
            description
              "This value describes the scope of vlan:
        use ',' or '-' to indicate the scope, e.g. 1,3,5-7.";
            }
          }
        }
      }
    }
    <CODE ENDS>
```

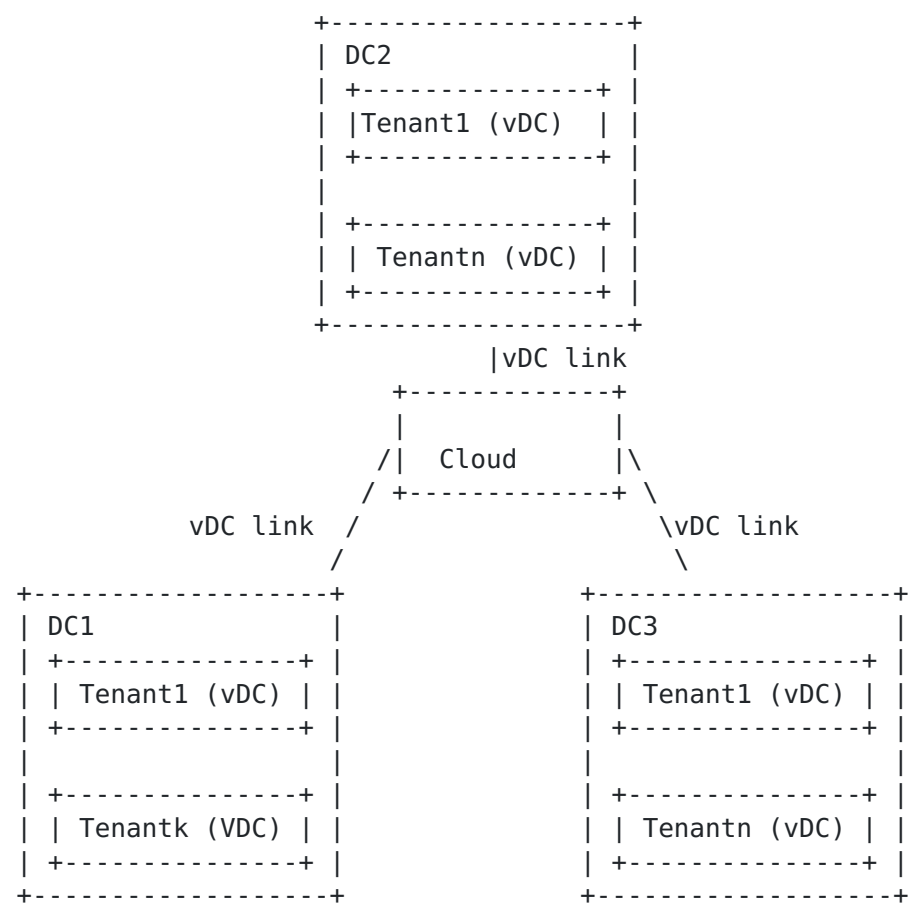
3.4. Module for DDC services

The following describes SUPA VPN management model designed for DDC services use case [SUPA-DDC]. [SUPA-DDC] took a large-scale Internet Data Center (IDC) operator as an example to describe what SUPA needs to do including DDC service initiation, VPN-based

connectivity initiation, optimize traffic route, traffic adjustment and monitor.

Module "ietf-supa-ddc" defines generic VPN management aspects which are common to all DDC services use case regardless of their type of vendor. In effect, the module can be viewed as providing a generic VPN management for DDC services.

This model is designed based on vDC Connectivity use case in section 6.2 of [SUPA-DDC]. The figure below showed the tenants distributed on multiple DCs and the connectivity among them. Service data model and policy data model can be defined to automate or simplify the links configuration for vDCs.



Please note that, for DDC model, it is essentially a data center service model, but with the help of L3VPN do bearing, so it refers to L3VPN business model (not inheritance relationships).

```
module: ietf-supa-ddc
   +--rw ddc-services
      +--rw ddc-service* [name]
         +--rw name               string
         +--rw tenant-name?       string
         +--rw dc-name*           string
         +--rw interface-name*    string
         +--rw connection-type?   enumeration
         +--rw connection-name    leafref
         +--rw bandwidth?         uint32
         +--rw latency?           uint32
```

3.4.1. Model for DDC services

```
<CODE BEGINS>
module ietf-supa-ddc {
  namespace "urn:ietf:params:xml:ns:yang:ietf-supa-ddc";
  // replace with IANA namespace when assigned
  prefix ddc;

  // import ietf-supa-vpn,ietf-supa-abstacted-l3vpn,
  // ietf-supa-l2vpn



  organization "IETF";
  contact
     "Editor: Ying Cheng
      chengying10@chinaunicom.cn

      Maxim Klyus
      klyus@NetCracker.com
     ";

  description
     "This YANG module defines a component that describing
      the ddc service model for creating and optimizing
      tenant's DC (data center) services that are deployed
      in multiple data centers.

      Terms and Acronyms
        DDC: Distributed Data Center
        L2VPN: Layer 2 Virtual Private Network
        L3VPN: Layer 3 Virtual Private Network
      ";
```

```
     revision 2014-12-25 {
        description
           "Initial revision.";
        reference "RFC4364, RFC7277";
     }

     container ddc-services {
       description
         "To create service for tenant's network that are deployed
         in multiple data centers. The following data are needed:
         name of data centers that the tenant's service are
         deployed in, connected method between data centers for
         the tenant (e.g. L2VPN, l3VPN, Native IP, etc.), name
         of tenant, ID of networks that belong to the tenant";

       list ddc-service {
         key "name";
         description
           "Overall ddc operational data, including the names of
           data center,the connection method between data centers,
           name of service, etc.";

         leaf name {
           type string;
           description
             "Indicates the name of the service";
         }

         leaf tenant-name {
           type string;
           description
             "Indicates the name of the tenant for whom
         the ddc service is being created.";
         }

         leaf-list dc-name {
           type string;
           description
             "List of the names of data center on which
             tenant's service is deployed in.";
         }

         leaf-list interface-name {
           type string;
           description
             "Indicates a set of access interface
             names of the network device that the
```

```
            data centers (deployment of tenant's
            service)are connected to.";
          }

          leaf connection-type {
            type enumeration {
              enum L2VPN {
                value 0;
                description "L2VPN";
              }
            enum L3VPN {
                value 1;
                description "L3VPN";
              }
              enum native-ipv4 {
                value 2;
                description "native IPv4";
              }
              enum native-ipv6 {
                value 3;
                description "native IPv6";
              }
            }
            description
              "Indicates the connection  type between the
              Data centers on which tenant service is being
              deployed . The connection type may be VPN
              (L2VPN or L3VPN) or Native IP (IPv4 or IPv6)";
          }

          leaf connection-name {
        type leafref { path "/l2vpn:l2vpn-instance/
          instance-name"; }
            mandatory true;
            description
              "Indicates the name of the connection e.g.,VPN
              instance";
          }

          leaf bandwidth {
            type uint32;
            description
              "Indicates the bandwidth of the network connection
              instance that is created for tenant.";
          }

          leaf latency {
```

```
          type uint32;
          description
            "Indicates the latency of the network connection
            instance that is created for tenant.";
        }
      }
    }
  }
  <CODE ENDS>
```

4. Annex A (Service instance common attributes)

The below code with set of common attributes can be implemented for each type of VPN service.

Attributes description:

Operational State - operability of a VPN service is described by the instance-oper-state read-only attribute, which has two possible values: disabled and enabled. Operational State changes can be driven by specific events associated with the VPN components and specific administrative VPN transitions.

instance-oper-state attribute states:
- enabled: service is enabled and can operate normally or with some errors
- disabled: service disabled administratively or automatically based on some critical errors

Administrative State - administration of managed VPN service described by instance-admin-state read-write attribute, which has two possible values: disabled and enabled. This attribute operates independently of the operability and usage of managed VPN and could be changed automatically based on specific events associated with VPN components or manually.

instance-admin-state attribute states:
- enabled: service administratively enabled, service configuration structure unchanged, but service component states can be changed)
- disabled: service administratively disabled, service configuration structure unchanged, but service component states can be changed)

Life Cycle State - planning and tracking of managed VPN service described by life-cycle-state read-write attribute. Inventoried VPN

components also may have a life cycle attribute so that their deployment can be planned, tracked and managed.

The life-cycle-state attribute shall have one of the following values
- planned: VPN is planned; configuration is ready but not installed in the network.
- installed: VPN service is ready for production; configuration is installed in the network.
- removed: The VPN service has been removed, configuration uninstalled from the network and saved for the history.

Availability Status - availability of a VPN service described by the instance-availability-status read-only attribute, which can be zero or has one of two possible values: degraded or failed. Availability Status changes can be driven by specific events associated with the VPN components and specific administrative VPN transitions.

instance-availability-status attribute states:
- degraded: VPN is still can operate but some of the VPN components have an errors.
- failed: VPN service can't operate properly; because some of the VPN components have critical errors and VPN Operational State seems to be disabled.

Configuration Revision - configuration management of a VPN service is described by the instance-cfg-revision read-write string attribute. Service configuration could be modified (expanded/reduced) and VPN service instance will be unchanged but service components can be changed. Based on this attribute configuration changes can be tracked for service modification history purpose and last active configuration can be obtained.

SLA Policy - SLA policy reference described by the instance-sla-policy read-write string attribute. Decision about VPN service Operational State (Enabled / Disabled) and Availability State (Degraded / Failed) can be very complicated and each SP can have own decision process and internal service integrity check implementation. SLA Policy attribute SP can implement some SLA decision policy per VPN group service basis (when different VPNs have similar SLA policy).

```
   <CODE BEGINS>

   leaf instance-oper-state {
     type enumeration {
       enum enabled {
         value 1;
         description
           "VPN Service is Enabled";
       }
       enum disabled {
         value 2;
         description
           "VPN Service is Disabled";
       }
     }
   }

   leaf instance-admin-state {
     type enumeration {
       enum enabled {
         value 1;
         description
           "VPN Service Administratively Enabled";
       }
       enum disabled {
         value 2;
         description
         "VPN Service Administratively Disabled";
       }
     }
   }

   leaf life-cycle-state {
     type enumeration {
       enum planned {
         value 1;
         description
           "VPN Service Planned";
       }
       enum installed {
         value 2;
         description
           "VPN Service Installed";
       }
       enum removed {
         value 3;
         description
```

```
            "VPN Service Removed";
        }
      }
    }

    leaf instance-availability-status {
      type enumeration {
        enum degraded {
          value 1;
          description
            "VPN Service Degraded";
        }
        enum failed {
          value 2;
          description
            "VPN Service Failed";
        }
      }
    }

    leaf instance-cfg-revision {
      type string;
      description
          "Indicates current service configuration revision";
    }

    leaf instance-sla-policy {
      type string;
      description
        "Indicates SLA decision policy name";
    }

    <CODE ENDS>
```

5. Security Considerations

TBD

6. IANA Considerations

This document has no actions for IANA.

7. Contributors and Acknowledgments

This document has benefited from reviews, suggestions, comments and proposed text provided by the following members, listed in alphabetical order: Feng Dong, Jing Huang, Kepeng Li, Junru Lin, Felix Lu, Wu Nan, Juergen Schoenwaelder, Yiyong Zha, and Cathy Zhou.

Contributors:

Andy Bierman
andy@yumaworks.com

Juergen Schoenwaelder
j.schoenwaelder@jacobs-university.de

8. Additional Author List

The following are extended authors who contributed to the effort:

Kostas Pentikousis
EICT GmbH
Torgauer Strasse 12-15
Berlin 10829
Germany
Email: k.pentikousis@eict.de

Vikram Choudhary
Huawei
Email: vikram.choudhary@huawei.com

Will Liu
Huawei
Email: liushucheng@huawei.com

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

[RFC6021] Schoenwaelder, J., "Common YANG Data Types", RFC 6021, October 2010.

[RFC4110] Callon, R. and M. Suzuki, "A Framework for Layer 3 Provider-Provisioned Virtual Private Networks (PPVPNs)", RFC 4110, July 2005.

[RFC3272] Awduche, D., Chiu, A., Elwalid, A., Widjaja, I., and X. Xiao, "Overview and Principles of Internet Traffic Engineering", RFC 3272, May 2002.

[RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, December 1998.

[RFC4664] L. Andersson and E. Rosen, "Framework for Layer 2 Virtual Private Networks (L2VPNs)", RFC 4664, September 2006.

9.2. Informative References

[SUPA-framework] C. Zhou, L. M. Contreras, Q. Sun, and P. Yegani, " The Framework of Simplified Use of Policy Abstractions (SUPA) ", IETF Internet draft, draft-zhou-supa-framework, February 2015.

[SUPA-problem-statement] G. Karagiannis, Q. Sun, Luis M. Contreras, P. Yegani, JF Tremblay and J. Bi, "Problem Statement for Simplified Use of Policy Abstractions (SUPA)", IETF Internet draft, draft-karagiannis-supa-problem-statement, January 2015.

[SUPA-DDC] Y. Cheng, and JF. Tremblay, "Use Cases for Distributed Data Center Applications in SUPA", IETF Internet draft, draft-cheng-supa-ddc-use-cases, January 2015

[RESTCONF] Bierman, A., Bjorklund, M., Watsen, K., and R. Fernando, "RESTCONF Protocol", draft-ietf-netconf-restconf (work in progress), July 2014.

[VPN] Mason, Andrew G. (2002). Cisco Secure Virtual Private Network. Cisco Press. p. 7.

[l3vpn-service-yang] S. Litkowski, R. Shakir, L. Tomotaki and K. D'Souza, "YANG Data Model for L3VPN service delivery", IETF Internet draft, draft-l3vpn-service-yang, February 2015.

Authors' Addresses

Dacheng Zhang

Chaoyang Dist
Beijing 100000
P.R. China
Dacheng.zhang@gmail.com

Adel Zaalouk
EICT GmbH
Torgauer Strasse 12-15
Berlin 10829
Germany
Email: adel.ietf@gmail.com

Ying Cheng (Editor)
China Unicom
P.R. China

Email: chengying10@chinaunicom.cn

Jan Lindblad
Tail-F

Email: janl@tail-f.com

Maxim Klyus
NetCracker

Email: klyus@NetCracker.com