L3SM Working Group S. Litkowski
Internet-Draft Orange Business Service
Intended status: Standards Track R. Shakir
Expires: September 12, 2016 BT
L. Tomotaki
Verizon
K. Ogaki
KDDI
K. D'Souza
ATT

March 11, 2016

YANG Data Model for L3VPN service delivery
draft-ietf-l3sm-l3vpn-service-model-05

Abstract

This document defines a YANG data model that can be used to deliver a Layer 3 Provider Provisioned VPN service. The document is limited to the BGP PE-based VPNs as described in RFC4110 and RFC4364. This model is intended to be instantiated at management system to deliver the overall service. This model is not a configuration model to be used directly on network elements. This model provides an abstracted view of the Layer 3 IPVPN service configuration components. It will be up to a management system to take this as an input and use specific configurations models to configure the different network elements to deliver the service. How configuration of network elements is done is out of scope of the document.

Requirements Language

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].

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 September 12, 2016.

Table of Contents

1. Introduction 3
1.1. Terminology 3
1.2. Tree diagram 4
2. Definitions 4
3. Layer 3 IP VPN service model 5
4. Service data model usage 5
5. Design of the Data Model 6
5.1. VPN service overview 14
5.1.1. VPN service topology 15
5.1.1.1. Route Target allocation 15
5.1.1.2. Any to any 16
5.1.1.3. Hub and Spoke 16
5.1.1.4. Hub and Spoke disjoint 17
5.1.2. Cloud access 18
5.1.3. Multicast service . . . . . .

5.1.4. Extranet VPNs 20
5.2. Site overview 21
5.2.1. Site role 22
5.2.2. Site belonging to multiple VPNs 23
5.2.2.1. Site vpn flavor 24
5.2.2.2. Attaching a site to a VPN 24
5.2.3. Security 26
5.2.3.1. Encryption 29
5.2.4. Management 30
5.2.5. Routing protocols 30
5.2.5.1. Dual stack handling 30

5.2.5.2. Direct LAN connection onto SP network 31
5.2.5.3. Direct LAN connection onto SP network with redundancy 32
5.2.5.4. Static routing 32
5.2.5.5. RIP routing 32
5.2.5.6. OSPF routing 32
5.2.5.7. BGP routing 34
5.2.6. Service 36
5.2.6.1. QoS 36
5.2.6.2. Multicast 41
5.2.7. Site network accesses 41
5.2.7.1. Bearer 41
5.2.7.2. Connection 41
5.2.8. Deciding where to connect the site 42
5.2.8.1. Site location 42
5.2.8.2. Site diversity 43
5.2.8.3. Site network access availability 44
5.2.8.4. Route Distinguisher and VRF allocation 46
5.3. Enhanced VPN features 47
5.3.1. Carrier Supporting Carrier 47
5.3.2. Transport constraints 49
5.4. Using configuration templates 49
6. Service model usage example 54
7. Interaction with Other YANG Modules 58
8. YANG Module 63
9. Security Considerations 108
10. Acknowledgements . . . . . .

108
11. IANA Considerations 108
12. References 108
12.1. Normative References 108
12.2. Informative References 109
Appendix A. Example: NETCONF <get> Reply 109
Authors' Addresses 109

1. Introduction

This document defines a YANG data model for Layer 3 IPVPN service configuration.

1.1. Terminology

The following terms are defined in [RFC6241] and are not redefined here:

- client
- configuration data

- server
- state data

The following terms are defined in [RFC6020] and are not redefined here:

- augment
- data model
- data node

The terminology for describing YANG data models is found in [RFC6020].

1.2. Tree diagram

A simplified graphical representation of the data model is presented in Section 5.

The meaning of the symbols in these diagrams is as follows:

- Brackets "[" and "]" enclose list keys.
- Curly braces "{" and "}" contain names of optional features that make the corresponding node conditional.
- Abbreviations before data node names: "rw" means configuration (read-write), and "ro" state data (read-only).
- Symbols after data node names: "?" means an optional node and "*" denotes a "list" or "leaf-list".
- Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").
- Ellipsis ("...") stands for contents of subtrees that are not shown.

2. Definitions

Customer Edge (CE) Device: The equipment on the customer side of the SP-customer boundary (the customer interface).

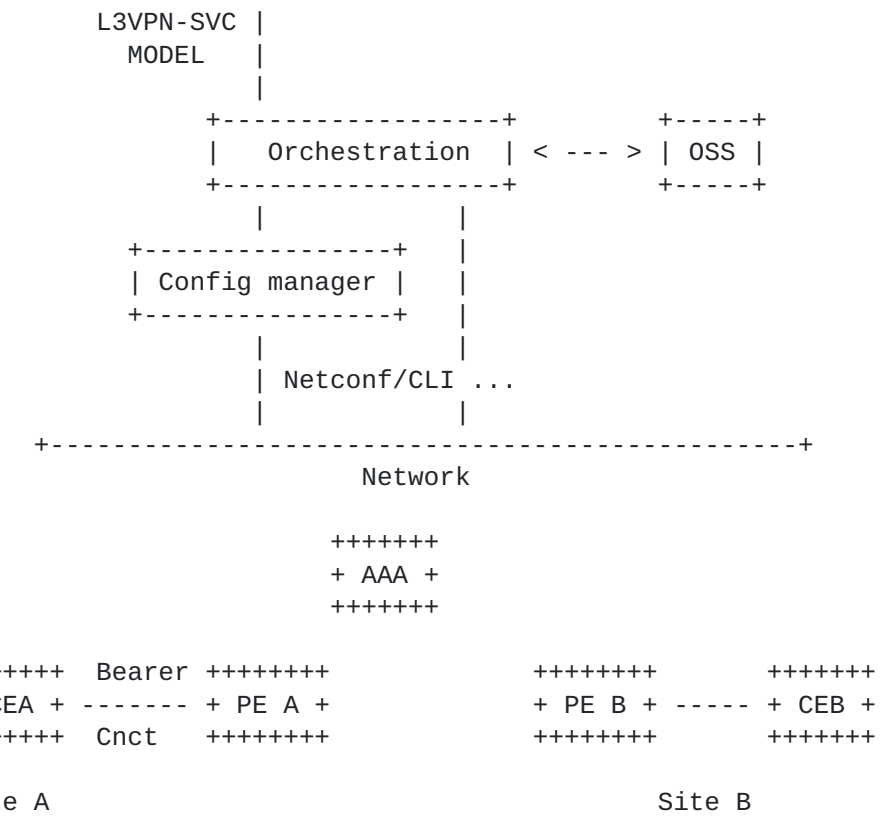
Provider Edge (PE) Device: The equipment on the SP side of the SP-customer boundary (the customer interface).

PE-Based VPNs: The PE devices know that certain traffic is VPN traffic. They forward the traffic (through tunnels) based on the destination IP address of the packet, and optionally on based on other information in the IP header of the packet. The PE devices are themselves the tunnel endpoints. The tunnels may make use of various encapsulations to send traffic over the SP network (such as, but not restricted to, GRE, IP-in-IP, IPsec, or MPLS tunnels).

3. Layer 3 IP VPN service model

A Layer 3 IPVPN service is a collection of sites that are authorized to exchange traffic between each other over a shared IP infrastructure. This layer 3 VPN service model aims at providing a common understanding on how the corresponding IP VPN service is to be deployed over the shared infrastructure. This service model is limited to BGP PE-Based VPNs as described in [RFC4110] and [RFC4364].

4. Service data model usage



The idea of the L3 IPVPN service model is to propose an abstracted interface to manage configuration of components of a L3VPN service. A typical usage is to use this model as an input for an orchestration layer who will be responsible to translate it to orchestrated configuration of network elements who will be part of the service. The network elements can be routers, but also servers (like AAA), and not limited to these examples. The configuration of network elements MAY be done by CLI, or by NetConf/RestConf coupled with specific configuration YANG data models (BGP, VRF, BFD ...) or any other way.

The usage of this service model is not limited to this example, it can be used by any component of the management system but not directly by network elements.

5. Design of the Data Model

The YANG module is divided in three main containers : vpn-svc, sites, site-templates.

The vpn-svc defines global parameters for the VPN service for a specific customer.

A site is composed of at least one site-network-access and may have multiple site-network-access in case of multihoming. The site-network-access attachment is done through a bearer with a connection (transport protocol) on top. The bearer refers to physical properties of the attachment while the connection refers to more protocol oriented properties.

Authorization of traffic exchange is done through what we call a VPN policy or VPN topology defining routing exchange rules between sites.

The site-templates may be used as configuration templates for sites. Part of the site configuration can be inherited from templates.

The figure below describe the overall structure of the YANG module:

```
module: ietf-l3vpn-svc
   +--rw l3vpn-svc
      +--rw vpn-services
      |  +--rw vpn-svc* [vpn-id]
      |     +--rw vpn-id                    svc-id
      |     +--rw customer-name?            string
      |     +--rw topology?                 identityref
      |     +--rw cloud-access* [cloud-identifier]
      |     |  +--rw cloud-identifier         string
      |     |  +--rw authorized-sites* [site-id]
```

```
      |     |  |  +--rw site-id    leafref
```

```
|     |  +--rw denied-sites* [site-id]
|     |  |  +--rw site-id    leafref
|     |  +--rw nat-enabled?            boolean
|     |  +--rw customer-nat-address?   inet:ipv4-address
|     +--rw multicast
|     |  +--rw enabled?                 boolean
|     |  +--rw customer-tree-flavors
|     |  |  +--rw tree-flavor* [type]
|     |  |     +--rw type    identityref
|     |  +--rw rp
|     |     +--rw rp-group-mapping* [rp-address group]
|     |     |  +--rw provider-managed
|     |     |  |  +--rw enabled?      boolean
|     |     |  |  +--rw anycast-rp?   boolean
|     |     |  +--rw rp-address          union
|     |     |  +--rw group               union
|     |     +--rw rp-discovery?       identityref
|     +--rw mpls?                    boolean
|     +--rw transport-constraints
|     |  +--rw unicast-transport-constraints
|     |  |  +--rw constraints* [constraint-id]
|     |  |     +--rw constraint-id      svc-id
|     |  |     +--rw site1?             svc-id
|     |  |     +--rw site2?             svc-id
|     |  |     +--rw constraint-list* [constraint-type]
|     |  |        +--rw constraint-type            identityref
|     |  |        +--rw constraint-opaque-value?   string
|     |  +--rw multicast-transport-constraints
|     |     +--rw constraints* [constraint-id]
|     |        +--rw constraint-id      svc-id
|     |        +--rw src-site?          svc-id
|     |        +--rw dst-site?          svc-id
|     |        +--rw constraint-list* [constraint-type]
|     |           +--rw constraint-type            identityref
|     |           +--rw constraint-opaque-value?   string
|     +--rw extranet-vpns
|        +--rw extranet-vpn* [vpn-id]
|           +--rw vpn-id              svc-id
|           +--rw local-sites-role?   identityref
+--rw sites
|  +--rw site* [site-id]
|     +--rw site-id                  svc-id
|     +--rw apply-template?          leafref
|     +--rw requested-site-start?    yang:date-and-time
|     +--rw requested-site-stop?     yang:date-and-time
|     +--rw actual-site-start?       yang:date-and-time
|     +--rw actual-site-stop?        yang:date-and-time
|     +--rw location
```

```
|     |  +--rw address?        string
|     |  +--rw zip-code?       string
|     |  +--rw city?           string
|     |  +--rw country-code?   string
|     +--rw site-diversity
|     |  +--rw type?         placement-diversity
|     |  +--rw site-group*   uint32
|     +--rw management
|     |  +--rw type?                   identityref
|     |  +--rw management-transport?   identityref
|     |  +--rw address?                union
|     +--rw vpn-policy-list
|     |  +--rw vpn-policy* [vpn-policy-id]
|     |     +--rw vpn-policy-id    svc-id
|     |     +--rw entries* [id]
|     |        +--rw id        svc-id
|     |        +--rw filter
|     |        |  +--rw (lan)?
|     |        |        ...
|     |        +--rw vpn
|     |           +--rw vpn-id       leafref
|     |           +--rw site-role    identityref
|     +--rw site-vpn-flavor?         identityref
|     +--rw maximum-routes
|     |  +--rw address-family* [af]
|     |     +--rw af                identityref
|     |     +--rw maximum-routes?   uint32
|     +--rw security
|     |  +--rw authentication
|     |  +--rw encryption
|     |     +--rw enabled?              boolean
|     |     +--rw layer?                enumeration
|     |     +--rw encryption-profile
|     |        +--rw (profile)?
|     |           +--:(provider-profile)
|     |           |     ...
|     |           +--:(customer-profile)
|     |                 ...
|     +--rw service
|     |  +--rw svc-input-bandwidth?    uint32
|     |  +--rw svc-output-bandwidth?   uint32
|     |  +--rw svc-mtu?                uint16
|     |  +--rw qos
|     |  |  +--rw qos-classification-policy
|     |  |  |  +--rw rules* [id]
|     |  |  |     +--rw id                uint16
|     |  |  |     +--rw match-flow
|     |  |  |     |     ...
```

```
|     |  |  |     +--rw target-class-id?   string
|     |  |  +--rw qos-profile
|     |  |     +--rw (qos-profile)?
|     |  |        +--:(standard)
|     |  |        |     ...
|     |  |        +--:(custom)
|     |  |              ...
|     |  +--rw mpls
|     |  |  +--rw signalling-type?   enumeration
|     |  +--rw multicast
|     |     +--rw multicast-site-type?           enumeration
|     |     +--rw multicast-transport-protocol
|     |     |  +--rw ipv4?   boolean
|     |     |  +--rw ipv6?   boolean
|     |     +--rw protocol-type?                 enumeration
|     +--rw routing-protocols
|     |  +--rw routing-protocol* [type]
|     |     +--rw type      identityref
|     |     +--rw ospf
|     |     |  +--rw address-family*   identityref
|     |     |  +--rw area-address?     yang:dotted-quad
|     |     |  +--rw metric?           uint16
|     |     |  +--rw sham-link* [target-site]
|     |     |     +--rw target-site    svc-id
|     |     |     +--rw metric?        uint16
|     |     +--rw bgp
|     |     |  +--rw autonomous-system?   uint32
|     |     |  +--rw address-family*      identityref
|     |     +--rw static
|     |     |  +--rw cascaded-lan-prefixes
|     |     |     +--rw ipv4-lan-prefixes* [lan next-hop]
|     |     |     |     ...
|     |     |     +--rw ipv6-lan-prefixes* [lan next-hop]
|     |     |           ...
|     |     +--rw rip
|     |     |  +--rw address-family*   identityref
|     |     +--rw vrrp
|     |        +--rw address-family*   identityref
|     +--rw site-network-accesses
|        +--rw site-network-access* [site-network-access-id]
|           +--rw site-network-access-id    svc-id
|           +--rw apply-template?           leafref
|           +--rw access-diversity
|           |  +--rw type?   placement-diversity
|           +--rw bearer
|           |  +--rw type?               string
|           |  +--rw bearer-reference?   string
|           +--rw ip-connection
```

```
      |           |  +--rw ipv4
      |           |  |  +--rw address-allocation-type?   identityref
      |           |  |  +--rw (subnet)?
      |           |  |        ...
      |           |  +--rw ipv6
      |           |  |  +--rw address-allocation-type?   string
      |           |  |  +--rw (subnet)?
      |           |  |        ...
      |           |  +--rw oam
      |           |     +--rw bfd
      |           |           ...
      |           +--rw security
      |           |  +--rw authentication
      |           |  +--rw encryption
      |           |     +--rw enabled?              boolean
      |           |     +--rw layer?                enumeration
      |           |     +--rw encryption-profile
      |           |           ...
      |           +--rw service
      |           |  +--rw svc-input-bandwidth?    uint32
      |           |  +--rw svc-output-bandwidth?   uint32
      |           |  +--rw svc-mtu?                uint16
      |           |  +--rw qos
      |           |  |  +--rw qos-classification-policy
      |           |  |  |     ...
      |           |  |  +--rw qos-profile
      |           |  |        ...
      |           |  +--rw mpls
      |           |  |  +--rw signalling-type?   enumeration
      |           |  +--rw multicast
      |           |     +--rw multicast-site-type?
enumeration
      |           |     +--rw multicast-transport-protocol
      |           |     |     ...
      |           |     +--rw protocol-type?
enumeration
      |           +--rw routing-protocols
      |           |  +--rw routing-protocol* [type]
      |           |     +--rw type      identityref
      |           |     +--rw ospf
      |           |     |     ...
      |           |     +--rw bgp
      |           |     |     ...
      |           |     +--rw static
      |           |     |     ...
      |           |     +--rw rip
      |           |     |     ...
      |           |     +--rw vrrp
      |           |           ...
      |           +--rw availability
```

```
      |           |  +--rw traffic-protection
      |           |  |  +--rw enabled?   boolean
      |           |  +--rw access-priority?      uint32
      |           +--rw vpn-attachment
      |              +--rw (attachment-flavor)
      |                 +--:(vpn-policy-id)
      |                 |     ...
      |                 +--:(vpn-id)
      |                       ...
      +--rw site-templates
         +--rw site-template* [site-template-id]
            +--rw site-template-id        template-id
            +--rw requested-site-start?   yang:date-and-time
            +--rw requested-site-stop?    yang:date-and-time
            +--rw actual-site-start?      yang:date-and-time
            +--rw actual-site-stop?       yang:date-and-time
            +--rw location
            |  +--rw address?        string
            |  +--rw zip-code?       string
            |  +--rw city?           string
            |  +--rw country-code?   string
            +--rw site-diversity
            |  +--rw type?         placement-diversity
            |  +--rw site-group*   uint32
            +--rw management
            |  +--rw type?                   identityref
            |  +--rw management-transport?   identityref
            |  +--rw address?                union
            +--rw vpn-policy-list
            |  +--rw vpn-policy* [vpn-policy-id]
            |     +--rw vpn-policy-id    svc-id
            |     +--rw entries* [id]
            |        +--rw id        svc-id
            |        +--rw filter
            |        |  +--rw (lan)?
            |        |        ...
            |        +--rw vpn
            |           +--rw vpn-id       leafref
            |           +--rw site-role    identityref
            +--rw site-vpn-flavor?        identityref
            +--rw maximum-routes
            |  +--rw address-family* [af]
            |     +--rw af                identityref
            |     +--rw maximum-routes?   uint32
            +--rw security
            |  +--rw authentication
            |  +--rw encryption
            |     +--rw enabled?             boolean
```

```
|     +--rw layer?                enumeration
|     +--rw encryption-profile
|        +--rw (profile)?
|           +--:(provider-profile)
|           |     ...
|           +--:(customer-profile)
|                 ...
+--rw service
|  +--rw svc-input-bandwidth?    uint32
|  +--rw svc-output-bandwidth?   uint32
|  +--rw svc-mtu?                uint16
|  +--rw qos
|  |  +--rw qos-classification-policy
|  |  |  +--rw rules* [id]
|  |  |     +--rw id                 uint16
|  |  |     +--rw match-flow
|  |  |     |     ...
|  |  |     +--rw target-class-id?   string
|  |  +--rw qos-profile
|  |     +--rw (qos-profile)?
|  |        +--:(standard)
|  |        |     ...
|  |        +--:(custom)
|  |              ...
|  +--rw mpls
|  |  +--rw signalling-type?   enumeration
|  +--rw multicast
|     +--rw multicast-site-type?            enumeration
|     +--rw multicast-transport-protocol
|     |  +--rw ipv4?   boolean
|     |  +--rw ipv6?   boolean
|     +--rw protocol-type?                  enumeration
+--rw routing-protocols
|  +--rw routing-protocol* [type]
|     +--rw type      identityref
|     +--rw ospf
|     |  +--rw address-family*   identityref
|     |  +--rw area-address?     yang:dotted-quad
|     |  +--rw metric?           uint16
|     |  +--rw sham-link* [target-site]
|     |     +--rw target-site    svc-id
|     |     +--rw metric?        uint16
|     +--rw bgp
|     |  +--rw autonomous-system?   uint32
|     |  +--rw address-family*      identityref
|     +--rw static
|     |  +--rw cascaded-lan-prefixes
|     |     +--rw ipv4-lan-prefixes* [lan next-hop]
```

```
|     |     |     ...
|     |     +--rw ipv6-lan-prefixes* [lan next-hop]
|     |           ...
|     +--rw rip
|     |  +--rw address-family*   identityref
|     +--rw vrrp
|        +--rw address-family*   identityref
+--rw site-network-access
   +--rw access-diversity
   |  +--rw type?   placement-diversity
   +--rw bearer
   |  +--rw type?               string
   |  +--rw bearer-reference?   string
   +--rw ip-connection
   |  +--rw ipv4
   |  |  +--rw address-allocation-type?   identityref
   |  |  +--rw (subnet)?
   |  |     +--:(subnet-only)
   |  |     |     ...
   |  |     +--:(addresses)
   |  |           ...
   |  +--rw ipv6
   |  |  +--rw address-allocation-type?   string
   |  |  +--rw (subnet)?
   |  |     +--:(subnet-only)
   |  |     |     ...
   |  |     +--:(addresses)
   |  |           ...
   |  +--rw oam
   |     +--rw bfd
   |        +--rw bfd-enabled?    boolean
   |        +--rw (holdtime)?
   |              ...
   +--rw security
   |  +--rw authentication
   |  +--rw encryption
   |     +--rw enabled?              boolean
   |     +--rw layer?                enumeration
   |     +--rw encryption-profile
   |        +--rw (profile)?
   |              ...
   +--rw service
   |  +--rw svc-input-bandwidth?    uint32
   |  +--rw svc-output-bandwidth?   uint32
   |  +--rw svc-mtu?                uint16
   |  +--rw qos
   |  |  +--rw qos-classification-policy
   |  |  |  +--rw rules* [id]
```

```
|  |  |        ...
|  |  +--rw qos-profile
|  |     +--rw (qos-profile)?
|  |           ...
|  +--rw mpls
|  |  +--rw signalling-type?   enumeration
|  +--rw multicast
|     +--rw multicast-site-type?            enumeration
|     +--rw multicast-transport-protocol
|     |  +--rw ipv4?   boolean
|     |  +--rw ipv6?   boolean
|     +--rw protocol-type?                  enumeration
+--rw routing-protocols
|  +--rw routing-protocol* [type]
|     +--rw type      identityref
|     +--rw ospf
|     |  +--rw address-family*   identityref
|     |  +--rw area-address?     yang:dotted-quad
|     |  +--rw metric?           uint16
|     |  +--rw sham-link* [target-site]
|     |        ...
|     +--rw bgp
|     |  +--rw autonomous-system?   uint32
|     |  +--rw address-family*      identityref
|     +--rw static
|     |  +--rw cascaded-lan-prefixes
|     |        ...
|     +--rw rip
|     |  +--rw address-family*   identityref
|     +--rw vrrp
|        +--rw address-family*   identityref
+--rw availability
|  +--rw traffic-protection
|  |  +--rw enabled?   boolean
|  +--rw access-priority?      uint32
+--rw vpn-attachment
   +--rw (attachment-flavor)
      +--:(vpn-policy-id)
      |  +--rw vpn-policy-id?   leafref
      +--:(vpn-id)
         +--rw vpn-id?          leafref
         +--rw site-role        identityref
```

5.1. VPN service overview

The vpn-svc top container contains generic information about the VPN service. The vpn-id of the vpn-svc refers to an internal reference for this VPN service, while customer name refers to a more explicit

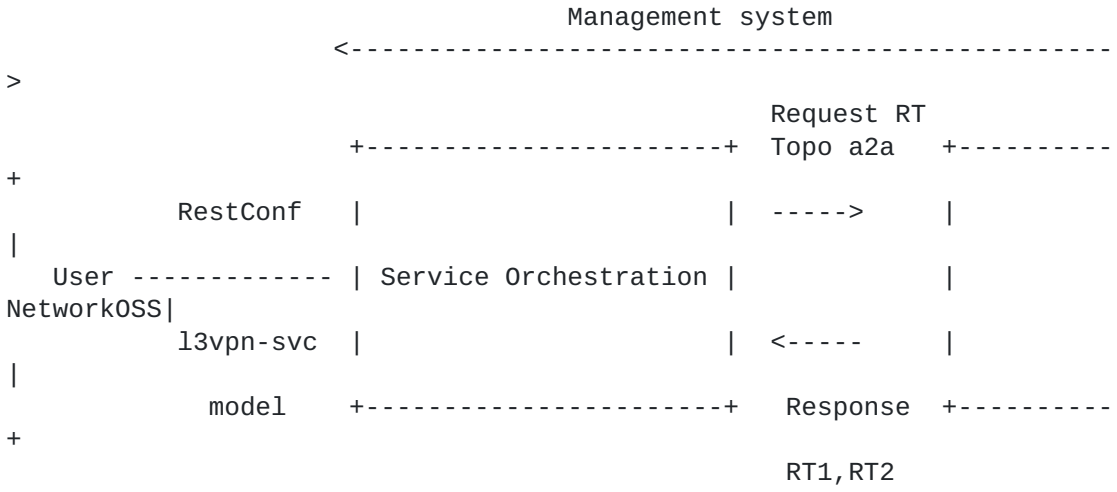
reference to the customer. This identifier is purely internal to the organization responsible for the VPN service. The vpn-id MUST be unique.

5.1.1. VPN service topology

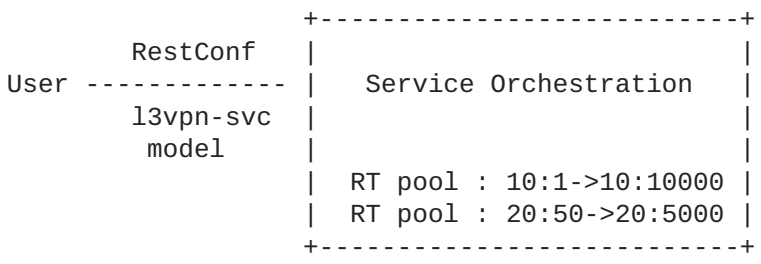
The type of topology of the VPN is required for configuration. Current proposal supports : any-to-any, hub and spoke (where hubs can exchange traffic), and hub and spoke disjoint (where hubs cannot exchange traffic). New topologies could be added by augmentation. By default, any-to-any topology is used.

5.1.1.1. Route Target allocation

Layer 3 PE-based VPN is built using route-targets as described in [RFC4364]. It is expected management system to allocate automatically set of route-targets upon a VPN service creation request. How management system allocates route-targets is out of scope of the document but multiple ways could be envisaged as described below.



In the example above, a service orchestration, owning the instantiation of this service model, request route-targets to the network OSS. Based on the requested VPN topology, the network OSS replies with one or multiple route-targets. The interface between this service orchestration and network OSS is out of scope of this document.



In the example above, a service orchestration, owning the instantiation of this service model, owns one or more pools of route-target (filled by service provider) that can be allocated. Based on the requested VPN topology, it will allocate one or multiple route-targets from the pool.

The mechanism displayed above are just examples and SHOULD NOT be considered as exhaustive list of solutions.

5.1.1.2. Any to any

Figure - Any to any VPN topology

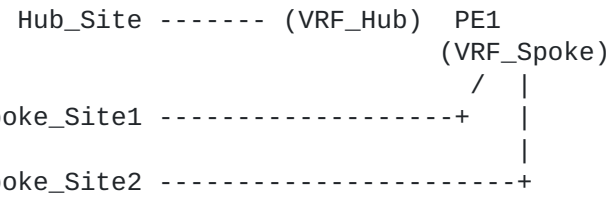
In the any to any topology, all VPN sites can discuss between each other without any restriction. It is expected that the management system that owns a any to any IPVPN service request through this model, needs to assign and then configure the VRF and route-targets on the appropriate PEs. In case of any to any, in general a single route-target is required and every VRF imports and exports this route-target.

5.1.1.3. Hub and Spoke

Figure - Hub and Spoke VPN topology

In the hub and spoke topology, all spoke sites can discuss only with Hub sites but not between each other. Hubs can discuss also between each other. It is expected that the management system that owns a any to any IPVPN service request through this model, needs to assign and then configure the VRF and route-targets on the appropriate PEs. In case of hub and spoke, in general a two route-targets are required (one route-target for Hub routes, one route-target for spoke routes). A Hub VRF, connecting Hub sites, will export Hub routes with Hub route-target, and will import Spoke routes through Spoke route-target. It will also import the Hub route-target to allow Hub to Hub communication. A Spoke VRF, connecting Spoke sites, will export Spoke routes with Spoke route-target, and will import Hub routes through Hub route-target.

The management system MUST take into account Hub and Spoke connections constraints. For example, if management system decides to mesh a spoke site and a hub site on the same PE, it needs to mesh connections in different VRFs as displayed in the figure below.



5.1.1.4. Hub and Spoke disjoint

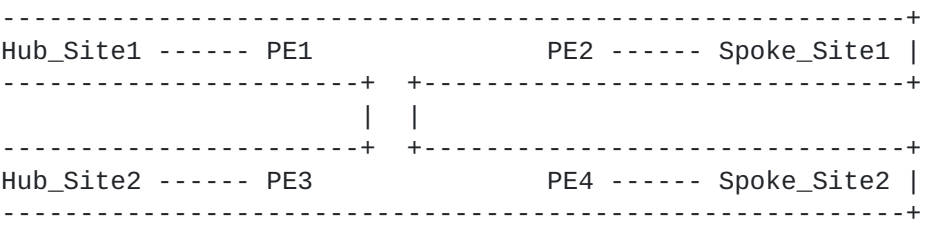


Figure - Hub and Spoke disjoint VPN topology

In the hub and spoke disjoint topology, all spoke sites can discuss only with Hub sites but not between each other. Hubs cannot discuss between each other. It is expected that the management system that owns a any to any IPVPN service request through this model, needs to assign and then configure the VRF and route-targets on the appropriate PEs. In case of hub and spoke, in general a two route-targets are required (one route-target for Hub routes, one route-target for spoke routes). A Hub VRF, connecting Hub sites, will export Hub routes with Hub route-target, and will import Spoke routes

through Spoke route-target. A Spoke VRF, connecting Spoke sites, will export Spoke routes with Spoke route-target, and will import Hub routes through Hub route-target.

The management system MUST take into account Hub and Spoke connections constraints as in the previous case.

5.1.2. Cloud access

The proposed model provides cloud access configuration through the cloud-access container. Internet access can typically be considered as a public cloud access service. The cloud-access container provides parameters for network address translation and authorization rules.

A cloud identifier is used to reference the target service. This identifier is local to each administration.

If NAT is required to access to the cloud, the nat-enabled leaf MUST be set to true. A NAT address may be provided in customer-nat-address, in case the customer is providing the public IP address for the cloud access. If service provider is providing the NAT address, customer-nat-address is not necessary as it can be picked from a service provider pool.

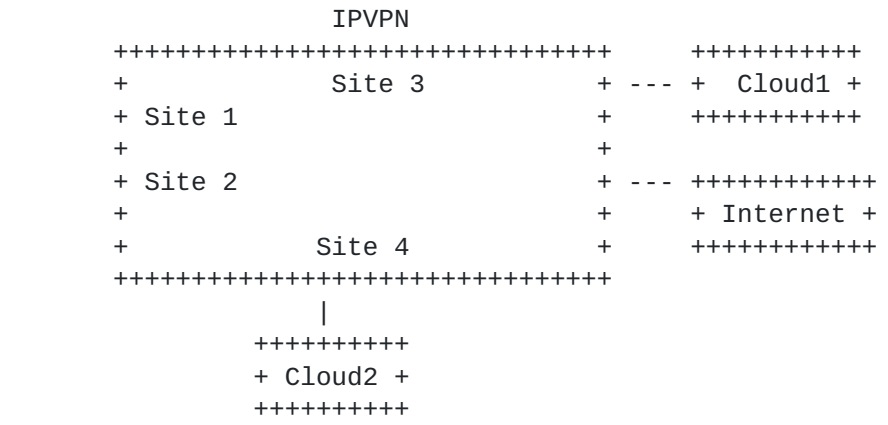
By default, all sites in the IPVPN MUST be authorized to access to the cloud. In case restrictions are required, a user MAY configure the authorized-sites and denied-sites list. The authorization-sites defines the list of sites authorized for cloud access. The denied-sites defines the list of sites denied for cloud access. The model supports both "deny all expect" and "accept all expect" authorization.

The "deny all expect" behavior is obtained by filling only the authorized-sites. All the sites listed will be authorized, all others will be denied.

The "accept all expect" behavior is obtained by filling only the denied-sites. All the sites listed will be denied, all others will be authorized.

Defining both denied-sites and authorized-sites MUST be processed as "deny all expect", so the denied-sites will have not effect.

How the restrictions will be configured on network elements is out of scope of this document and will be specific to each deployment.



In the example above, we may configure the global VPN to access Internet by creating a cloud-access pointing to the cloud identifier for Internet service. No authorized-sites will be configured as all sites are required to access to Internet. NAT-enabled will be set to true and a nat-address will be configured.

```
<vpn-svc>
    <vpn-id>ZKITYHJ054687</vpn-id>
    <customer-name>CUSTOMER_1</customer-name>
    <topology>any-to-any</topology>
    <cloud-access>
        <cloud-identifier>51</cloud-identifier>
        <nat-enabled>true</nat-enabled>
    </cloud-access>
</vpn-svc>
```

If Site1 and Site2 requires access to Cloud1, a new cloud-access will be created pointing to the cloud identifier of Cloud1. Authorized sites will be filled with reference to Site1 and Site2.

```
<vpn-svc>
    <vpn-id>12456487</vpn-id>
    <customer-name>CUSTOMER_1</customer-name>
    <topology>any-to-any</topology>
    <cloud-access>
        <cloud-identifier>1111111</cloud-identifier>
        <authorized-sites>
            <site-id>site1</site-id>
            <site-id>site2</site-id>
        </authorized-sites>
    </cloud-access>
</vpn-svc>
```

If all sites except Site1 requires access to Cloud2, a new cloud-access will be created pointing to the cloud identifier of Cloud2. denied-sites will be filled with reference to Site1.

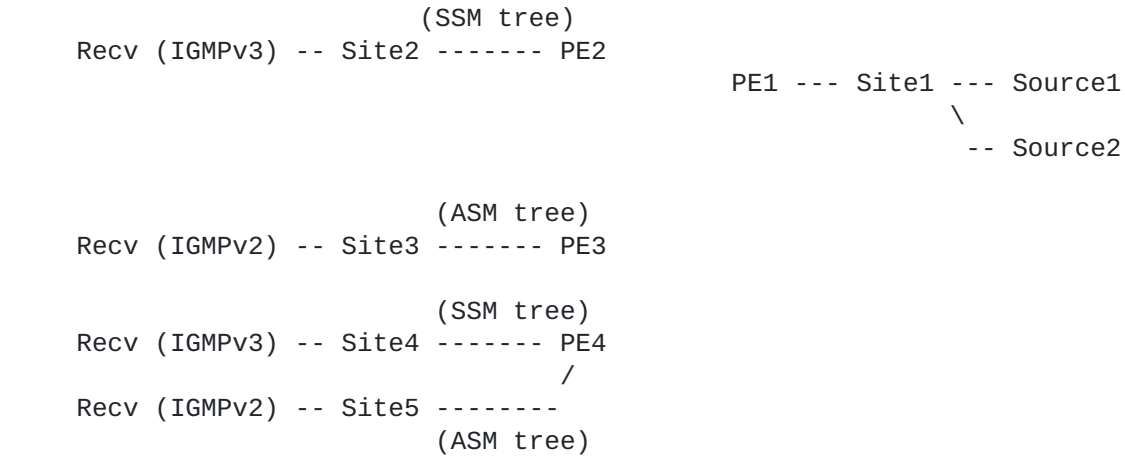
```
<vpn-svc>
    <vpn-id>12456487</vpn-id>
    <customer-name>CUSTOMER_1</customer-name>
    <topology>any-to-any</topology>
    <cloud-access>
        <cloud-identifier>22222222</cloud-identifier>
        <denied-sites>
            <site-id>site1</site-id>
        </denied-sites>
    </cloud-access>
</vpn-svc>
```

5.1.3. Multicast service

Multicast in IP VPN is described in [RFC6513].

If IPVPN supports multicast service, it is expected to provide inputs on global multicast parameters.

The user of this model will need to fill the flavor of trees that will be used by customer within the IPVPN (Customer tree). The proposed model supports ASM, SSM and BiDirectional trees (and can be augmented). Multiple flavors of tree can be supported simultaneously.



In case of ASM flavor, this model requires to fill the rp and rp-discovery parameters. Multiple RP to group mappings can be created. The RP service can be managed by the service provider using the leaf

provider-managed/enabled set to true. In case of provider managed RP, anycast RP can also be activated providing redundancy as well as more optimal forwarding.

In case of customer managed RP, the RP address must be filled in the RP to group mappings.

The rp-discovery supports the auto-rp, static-rp, anycast-rp and bsr-rp modes.

5.1.4. Extranet VPNs

There are some cases where a particular VPN needs to access to resources that are external. The resources may be located in another VPN.

In the figure above, VPN B has some resources on Site B that need to be available to some customers/partners. VPN A must be able to access those VPN B resources.

Such VPN connection scenario can be achieved by the VPN policy defined in Section 5.2.2.2.2. But there are some simple cases, where a particular VPN (VPN A) needs to access to all resources in a VPN B. The model provides an easy way to setup this connection using the extranet-vpns container.

The extranet-vpns container defines a list of VPNs, a particular VPN wants to access. The extranet-vpns must be used on "customer" VPNs accessing extranet resources in another VPN. In the figure above, in order to give access for VPN A to VPN B, extranet-vpns container will be configured with an entry corresponding to VPN B. An implementation MUST consider the relation as bi-directional : if A requires access to B, the derived device configuration must let B be aware of A.

The site-role leaf defines the role of the local VPN sites in the target extranet VPN topology. Site roles are defined in Section 5.2.1.

In the example below, VPN A accesses to VPN B resources through extranet connection, a spoke role is required for VPN A sites, so sites from VPN A must not be able to communicate between each other through the extranet VPN connection.

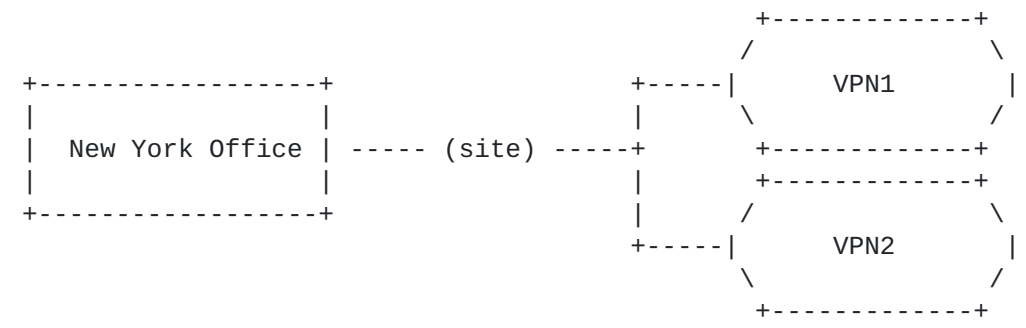
```
<vpn-svc>
    <vpn-id>VPNB</vpn-id>
    <topology>hub-spoke</topology>
</vpn-svc>
<vpn-svc>
    <vpn-id>VPNA</vpn-id>
    <topology>any-to-any</topology>
    <extranet-vpns>
        <extranet-vpn>
            <vpn-id>VPNB</vpn-id>
            <site-role>spoke-role</site-role>
        </extranet-vpn>
    </extranet-vpns>
</vpn-svc>
```

This model does not define how the extranet configuration will be achieved.

Any more complex VPN connection topology (e.g. only part of sites of VPN A accessing only part of sites of VPN B) needs to be achieved using the vpn attachment defined in Section 5.2.2.2.

5.2. Site overview

A site represents a connection of a customer location to one or more VPN services.



A site is composed of some characteristics :

o Unique identifier (site-id) : to uniquely identify the site within the overall network infrastructure. The identifier is a string

allowing to any encoding for the local administration of the VPN service.

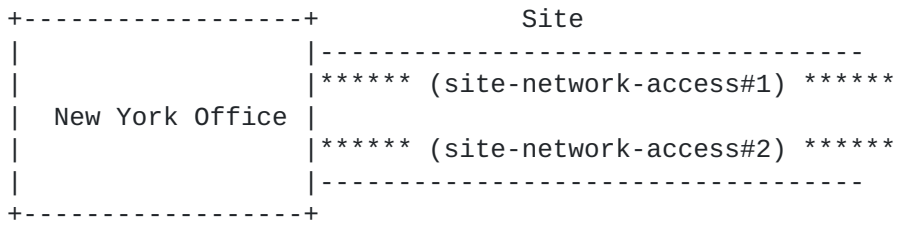
- Location (location) : site location informations to allow easy retrieval on nearest available resources.

- Site constraints (site-diversity) : site-diversity container allow to define some constraints for the setup of the site, for example : PE disjointness or PoP disjointness. A site-group identifier allow to manage the disjointness. Two sites with the same group and requiring PE disjointness cannot be connected on the same PE.

- Management (management) : defines the model of management of the site, for example : co-managed, customer managed or provider managed.

- Site network accesses (site-network-accesses) : defines the list of network accesses associated to the sites and their properties : especially bearer, connection and service parameters.

A site-network-access represents an IP logical connection of a site. A site may have multiple site-network-accesses.



Multiple site-network-accesses are used in case of multihoming.

The site configuration is viewed as a global entity, we assume that it is mostly the role of the management to split the parameters between the different elements within the network. For example, in the case of the site-network-access configuration, the management system needs to split the overall parameters between PE configuration and CE configuration.

5.2.1. Site role

A VPN has a particular topology as described in Section 5.1.1. As a consequence, each site belonging to a VPN as a particular role in this topology. The site-role defines the role of the site in a particular VPN topology.

In the any-to-any topology, all sites MUST have the same role which is any-to-any-role.

In the hub-spoke or hub-spoke-disjoint topology, sites MUST have a hub-role or a spoke-role.

5.2.2. Site belonging to multiple VPNs

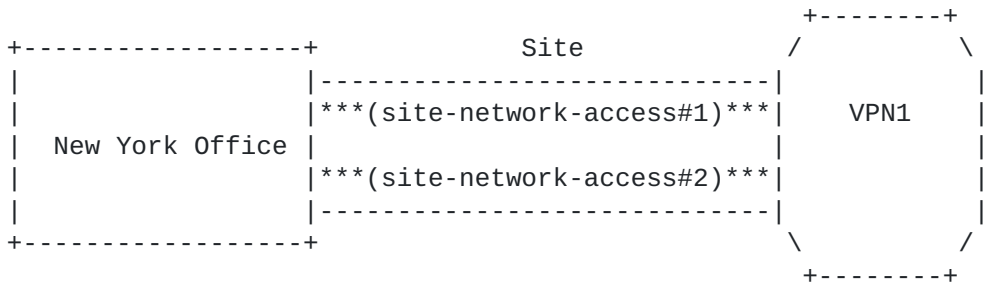
5.2.2.1. Site vpn flavor

A site may be part of one or multiple VPNs. The site flavor defines the way the VPN multiplexing is done. The current version of the model only supports two flavors :

- site-vpn-flavor-single : the site belongs to only one VPN.
- site-vpn-flavor-multi : the site belongs to multiple VPNs and all the logical accesses of the sites belongs to the same set of VPNs.
- site-vpn-flavor-sub : the site belongs to multiple VPNs with multiple logical accesses. Each logical access may map to different VPNs (one or many).

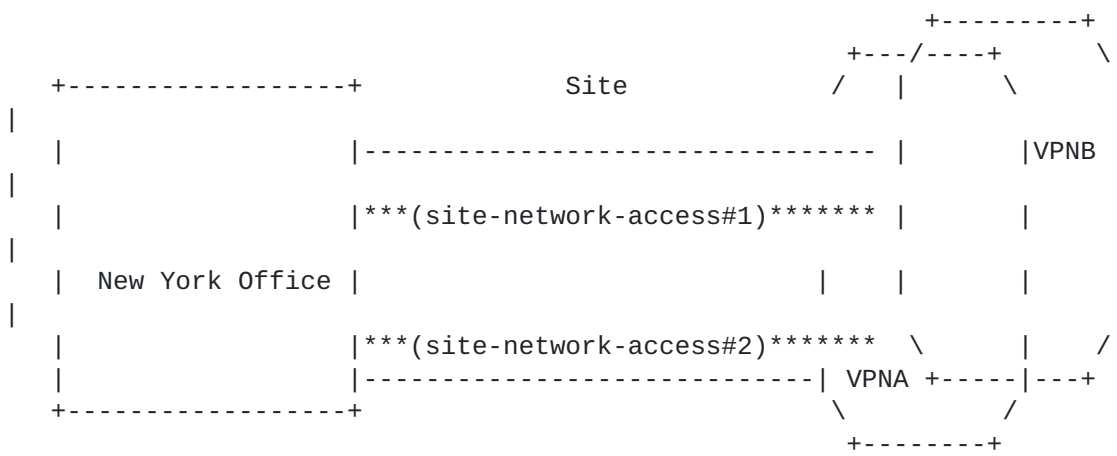
5.2.2.1.1. Single VPN attachment : site-vpn-flavor-single

The figure below describes the single VPN attachment. The site connects to only one VPN.



5.2.2.1.2. Multi VPN attachment : site-vpn-flavor-multi

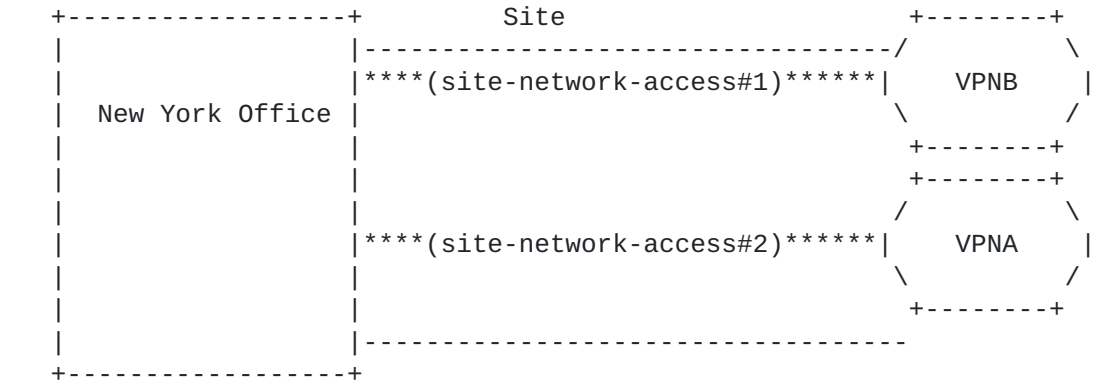
The figure below describes the multi VPN attachment. The site connects to multiple VPNs.



In the example above, the New York office is multihomed, both logical accesses are using the same VPN attachment rules. Both logical accesses are so connected to VPNA and VPNB.

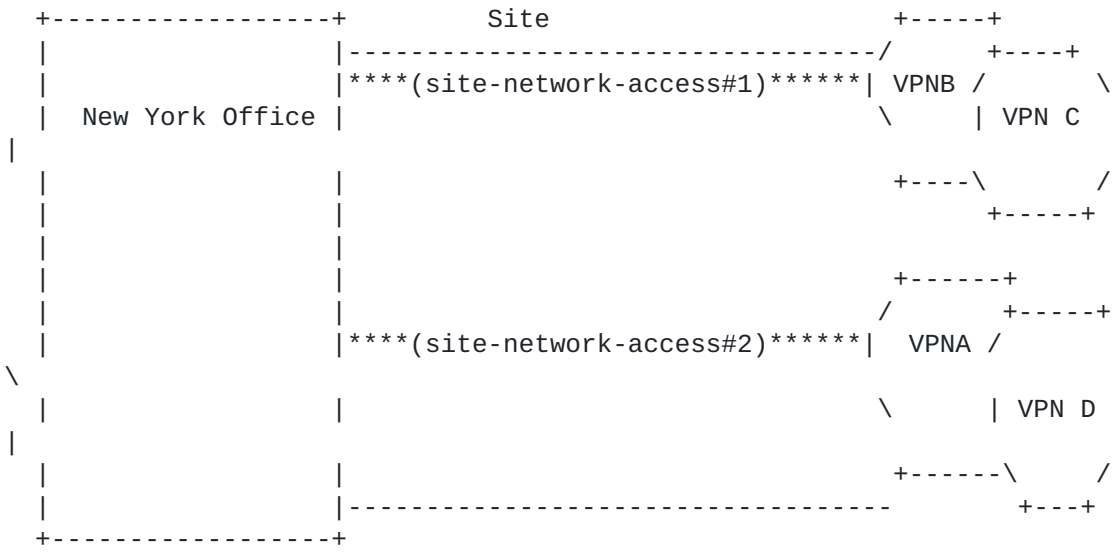
5.2.2.1.3. Sub VPN attachment : site-vpn-flavor-sub

The figure below describes a sub VPN attachment. The site connects to multiple VPNs but each logical access is attached to a particular set of VPN. Typical use case of subVPN is a customer site used by multiple affiliates with private resources for each affiliates that cannot be shared (communication is prevented between the affiliates).
It is similar than having separate sites instead that the customer wants to share some physical components while keeping strong isolation. In the example, the access#1 is attached to VPNB while the access#2 is attached to VPNA.

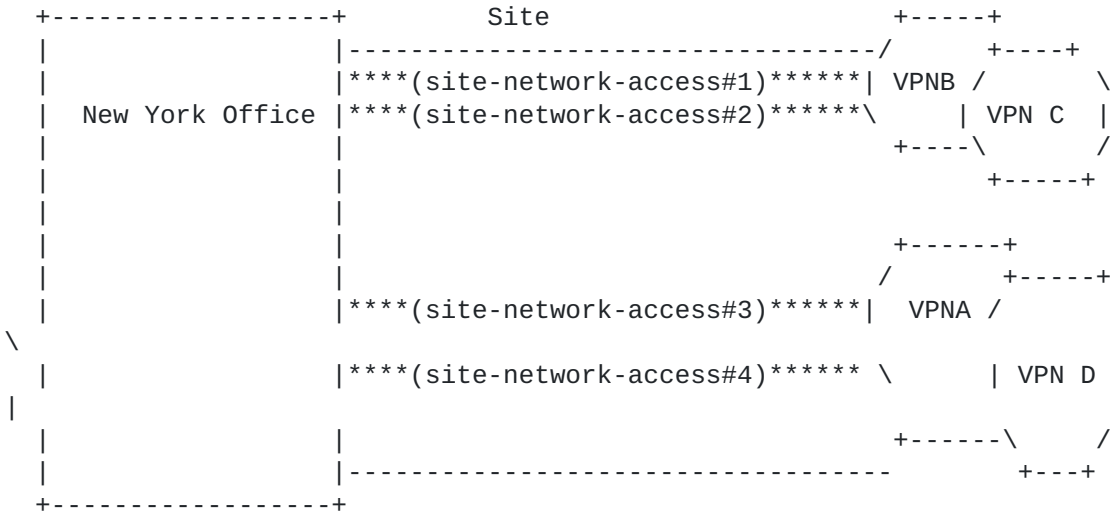


Multi-VPN can be implemented in addition to subVPN, as a consequence, each site-network-access can access to multiple VPNs. In the example

below, access#1 is mapped to VPNB and VPNC, while access#2 is mapped to VPNA and VPND.



Multihoming is also possible with subVPN, in this case, site-network-
accesses are grouped, and a particular group will access to the same set of VPN. In the example below, access#1 and #2 are part of the same group (multihomed together) and will be mapped to VPN B and C, in addition access#3 and #4 are part of the same group (multihomed together) and will be mapped to VPN A and D.



5.2.2.2. Attaching a site to a VPN

Due to the multiple site vpn flavors, the attachment is done at the site-network-access (logical access) level through the vpn-attachment
container. The vpn-attachment container is mandatory. The model provides two ways of attachment :

- Referencing directly the target VPN.

o Reference a VPN policy for more complex attachments.

A choice is implemented to allow user to choose the best fitting flavor.

5.2.2.2.1. Reference a VPN

Referencing a vpn-id provides an easy way to attach a particular logical access to a VPN. This is the best way in case of single VPN attachment or subVPN with single VPN attachment per logical access. When referencing a vpn-id, the site-role must be added to express the role of the site in the target VPN topology.

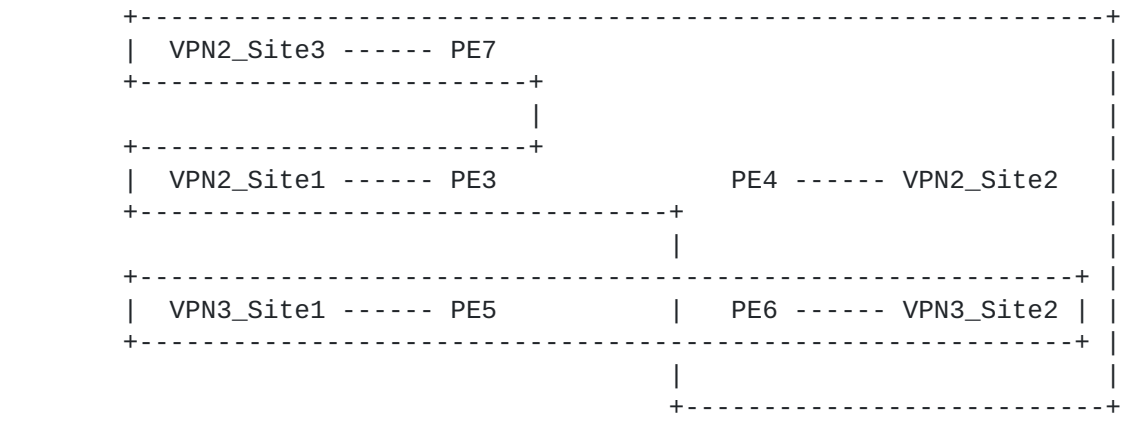
```
<site>
 <site-id>SITE1</site-id>
 <site-network-accesses>
  <site-network-access>
   <site-network-access-id>LA1</site-network-access-id>
   <vpn-attachment>
    <vpn-id>VPNA</vpn-id>
    <site-role>spoke-role</site-role>
   </vpn-attachment>
  </site-network-access>
  <site-network-access-id>LA2</site-network-access-id>
   <vpn-attachment>
    <vpn-id>VPNB</vpn-id>
    <site-role>spoke-role</site-role>
   </vpn-attachment>
  </site-network-access>
 </site-network-accesses>
</site>
```

The example above describes a subVPN case where a site SITE1 has two logical accesses (LA1 and LA2) with LA1 attached to VPNA and LA2 attached to VPNB.

5.2.2.2.2. VPN policy

The vpn-policy helps to express a multiVPN scenario where a logical access belongs to multiple VPNs. Multiple VPN policy can be created to handle the subVPN case where each logical access is part of a different set of VPNs.

As a site can belong to multiple VPNs, the vpn-policy may be composed of multiple entries. A filter can be applied to specify that only some LANs of the site should be part of a particular VPN. Each time a site (or LAN) is attached to a VPN, we must precise its role (site-role) within the targeted VPN topology.



In the example above, VPN3_Site2 is part of two VPNs : VPN3 and VPN2.
It will play hub-role in VPN2 and any-to-any role in VPN3. We can express such multiVPN scenario as follows :

```
<site>
 <site-id>VPN3_Site2</site-id>
 <vpn-policy-list>
  <vpn-policy>
   <vpn-policy-id>POLICY1</vpn-policy-id>
   <entries>
    <id>ENTRY1</id>
    <vpn>
     <vpn-id>VPN2</vpn-id>
     <site-role>hub-role</site-role>
    </vpn>
   </entries>
   <entries>
    <id>ENTRY2</id>
    <vpn>
     <vpn-id>VPN3</vpn-id>
     <site-role>any-to-any-role</site-role>
    </vpn>
   </entries>
  </vpn-policy>
 </vpn-policy-list>
 <site-network-accesses>
  <site-network-access>
   <site-network-access-id>LA1</site-network-access-id>
   <vpn-attachment>
    <vpn-policy-id>POLICY1</vpn-policy-id>
   </vpn-attachment>
  </site-network-access>
 </site-network-accesses>
</site>
```

Now in case more specific VPN attachment is necessary, filtering can be used. For example, if LAN1 from VPN3_site2 must be attached to VPN2 as hub and LAN2 must be attached to VPN3, the following configuration can be used :

```
<site>
 <site-id>VPN3_Site2</site-id>
 <vpn-policy-list>
  <vpn-policy>
   <vpn-policy-id>POLICY1</vpn-policy-id>
   <entries>
    <id>ENTRY1</id>
    <filter>
     <lan-tag>LAN1</lan-tag>
    </filter>
    <vpn>
     <vpn-id>VPN2</vpn-id>
     <site-role>hub-role</site-role>
    </vpn>
   </entries>
   <entries>
    <id>ENTRY2</id>
    <filter>
     <lan-tag>LAN2</lan-tag>
    </filter>
    <vpn>
     <vpn-id>VPN3</vpn-id>
     <site-role>any-to-any-role</site-role>
    </vpn>
   </entries>
  </vpn-policy>
 </vpn-policy-list>
 <site-network-accesses>
  <site-network-access>
   <site-network-access-id>LA1</site-network-access-id>
   <vpn-attachment>
    <vpn-policy-id>POLICY1</vpn-policy-id>
   </vpn-attachment>
  </site-network-access>
 </site-network-accesses>
</site>
```

5.2.3. Security

Security container defines customer specific security parameters for the site. This section will more be detailed in future revision.

5.2.3.1. Encryption

Encryption can be requested on the connection. It may be performed at layer 2 or layer 3 by selecting the appropriate enumeration in "layer" leaf. The encryption profile can be a service provider defined template or customer specific.

5.2.4. Management

The model proposes three types of common management options :

- comanaged : the CE router is managed by the provider and also by the customer.
- provider-managed : the CE router is managed only by the provider.
- customer-managed : the CE router is managed only by the customer.

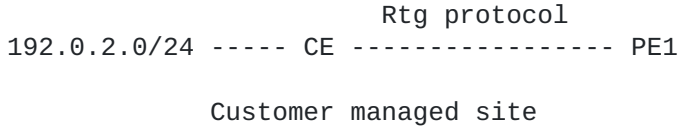
Based on the management model, different security options MAY be derived.

In case of "provider-managed" or "comanaged", the model proposes some option to define the management transport protocol (IPv4 or IPv6) and the associated management address.

5.2.5. Routing protocols

Routing-protocol defines which routing protocol must be activated between the provider and the customer router. The current model support : bgp, rip, rip-ng, ospf, static, direct, vrrp.

The routing protocol defined applies at the provider to customer boundary. Depending of the management of the management model, it may apply to the PE-CE boundary or CE to customer boundary. In case of customer managed site, the routing-protocol defined will be activated between the PE and the CE router managed by the customer. In case of provider managed site, the routing-protocol defined will be activated between the CE managed by the SP and the router or LAN belonging to the customer. In this case, it is expected that the PE-CE routing will be configured based on the service provider rules as both are managed by the same entity.



Customer managed site

```
            Rtg protocol
      Customer router ----- CE ----------------- PE1

                    Provider managed site
```

All the examples below will refer to a customer managed site case.

5.2.5.1. Dual stack handling

All routing protocol types support dual stack by using address-family
leaf-list.

Example of Dual stack using the same routing protocol :

```
<routing-protocols>
  <routing-protocol>
    <type>static</type>
    <static>
        <address-family>ipv4-unicast</address-family>
        <address-family>ipv6-unicast</address-family>
    </static>
  </routing-protocol>
</routing-protocols>
```

Example of Dual stack using two different routing protocols :

```
<routing-protocols>
  <routing-protocol>
    <type>rip</type>
    <rip>
        <address-family>ipv4-unicast</address-family>
    </rip>
  </routing-protocol>
  <routing-protocol>
    <type>ospf</type>
    <ospf>
        <address-family>ipv6-unicast</address-family>
    </ospf>
  </routing-protocol>
</routing-protocols>
```

5.2.5.2. Direct LAN connection onto SP network

Routing-protocol "direct" SHOULD be used when a customer LAN is
directly connected to the provider network and must be advertised in
the IPVPN.

LAN attached directly to provider network :

192.0.2.0/24 ----- PE1

5.2.5.3. Direct LAN connection onto SP network with redundancy

Routing-protocol "vrrp" SHOULD be used when a customer LAN is directly connected to the provider network and must be advertised in the IPVPN and LAN redundancy is expected.

LAN attached directly to provider network with LAN redundancy:

```
192.0.2.0/24 ------ PE1
                    |
                    +-- PE2
```

5.2.5.4. Static routing

Routing-protocol "static" MAY be used when a customer LAN is connected to the provider network through a CE router and must be advertised in the IPVPN.

```
                              Static rtg
192.0.2.0/24 ------ CE --------------- PE
                     |                 |
                     |      Static route 192.0.2.0/24 nh CE
     Static route 0.0.0.0/0 nh PE
```

5.2.5.5. RIP routing

Routing-protocol "rip" MAY be used when a customer LAN is connected to the provider network through a CE router and must be advertised in the IPVPN.

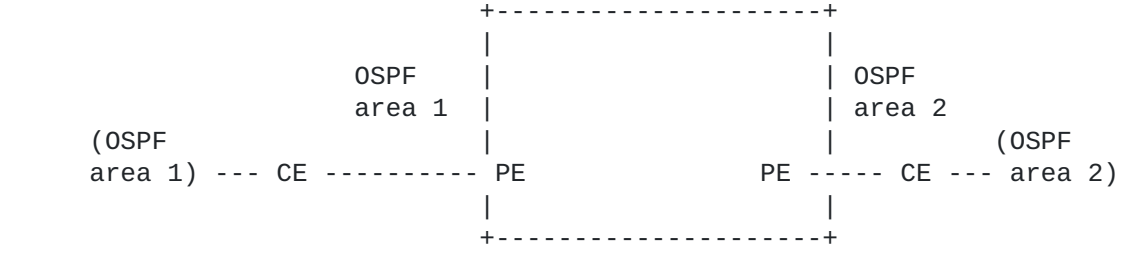
In case of dual stack, the management system will be responsible to configure rip (including right version number) and rip-ng instances on network elements.

```
                              RIP rtg
192.0.2.0/24 ------ CE --------------- PE
```

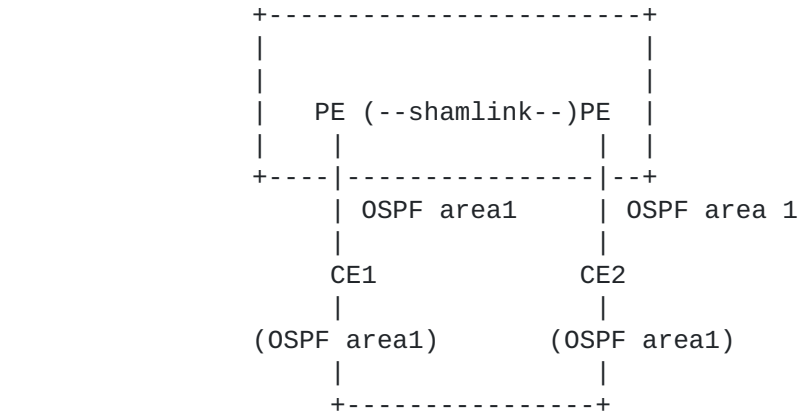
5.2.5.6. OSPF routing

Routing-protocol "ospf" MAY be used when a customer LAN is connected to the provider network through a CE router and must be advertised in the IPVPN.

It can be used to extend an existing OSPF network and interconnect different areas. See [RFC4577] for more details.



The model also proposes an option to create an OSPF sham-link between two sites sharing the same area and having a backdoor link. The sham-link is created by referencing the target site sharing the same OSPF area. The management system will be responsible to check if there is already a shamlink configured for this VPN and area between the same pair of PEs. If there is no existing shamlink, the management system will provision it, this shamlink MAY be reused by other sites.



Regarding Dual stack support, user MAY decide to fill both IPv4 and IPv6 address families, if both protocols SHOULD be routed through OSPF. As OSPF is using two different protocol for IPv4 and IPv6, the management system will need to configure both ospf version 2 and version 3 on the PE-CE link.

Example of OSPF routing parameters in service model.

```
   <routing-protocols>
     <routing-protocol>
       <type>ospf</type>
       <ospf>
           <area-address>0.0.0.1</area-address>
           <address-family>ipv4-unicast</address-family>
           <address-family>ipv6-unicast</address-family>
       </ospf>
     </routing-protocol>
   </routing-protocols>
```

Example of PE configuration done by management system :

```
router ospf 10
 area 0.0.0.1
  interface Ethernet0/0
!
router ospfv3 10
 area 0.0.0.1
  interface Ethernet0/0
 !
```

5.2.5.7. BGP routing

Routing-protocol "bgp" MAY be used when a customer LAN is connected to the provider network through a CE router and must be advertised in the IPVPN.

```
                                BGP rtg
      192.0.2.0/24 ------ CE -------------- PE
```

The session addressing will be derived from connection parameters as well as internal knowledge of SP.

In case of dual stack access, user MAY request BGP routing for both IPv4 and IPv6 by filling both address-families. It will be up to SP and management system to decide how to decline the configuration (two BGP sessions, single, multisession ...).

The service configuration below actives BGP on PE-CE link for both IPv4 and IPv6.

```
<routing-protocols>
  <routing-protocol>
    <type>bgp</type>
    <bgp>
        <autonomous-system>65000</autonomous-system>
        <address-family>ipv4-unicast</address-family>
        <address-family>ipv6-unicast</address-family>
    <bgp>
  </routing-protocol>
</routing-protocols>
```

This service configuration can be derived by management system into multiple flavors depending on SP flavor.

Example #1 of PE configuration done by management system (single session IPv4 transport):

```
router bgp 100
 neighbor 203.0.113.2 remote-as 65000
 address-family ipv4 vrf Cust1
    neighbor 203.0.113.2 activate
 address-family ipv6 vrf Cust1
    neighbor 203.0.113.2 activate
    neighbor 203.0.113.2 route-map SET-NH-IPV6 out
```

Example #2 of PE configuration done by management system (two sessions):

```
router bgp 100
 neighbor 203.0.113.2 remote-as 65000
 neighbor 2001::2 remote-as 65000
 address-family ipv4 vrf Cust1
    neighbor 203.0.113.2 activate
 address-family ipv6 vrf Cust1
    neighbor 2001::2 activate
```

Example #3 of PE configuration done by management system (multisession):

```
router bgp 100
 neighbor 203.0.113.2 remote-as 65000
 neighbor 203.0.113.2 multisession per-af
 address-family ipv4 vrf Cust1
    neighbor 203.0.113.2 activate
 address-family ipv6 vrf Cust1
    neighbor 203.0.113.2 activate
    neighbor 203.0.113.2 route-map SET-NH-IPV6 out
```

5.2.6. Service

The service defines service parameters associated with the site.

5.2.6.1. QoS

The model proposes to define QoS parameters in an abstracted way :

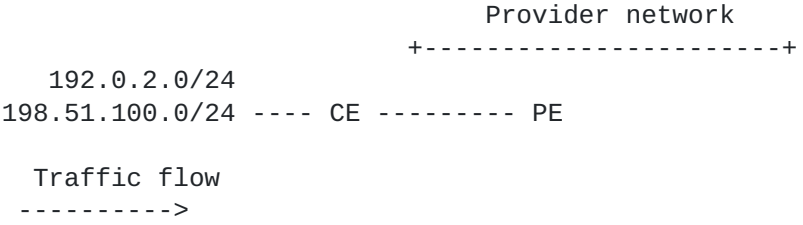
- qos-classification-policy : define a set of ordered rules to classify customer traffic.

- qos-profile : outgoing QoS profile to be applied.

5.2.6.1.1. QoS classification

QoS classification rules are handled by qos-classification-policy. The qos-classification-policy is an ordered list of rules that match a flow and set the appropriate target class of service (target-class-id). The match criterion provide a basic infrastructure for defining flows : layer 3 source and destination address, layer 4 ports, layer 4 protocol.

Where the classification is done depends on the SP implementation of the service, but classification concerns the flow coming from the customer site and entering the network.



In the figure above, the management system can decide :

- if the CE is customer managed, to implement the classification rule in the ingress direction on the PE interface.

- if the CE is provider managed, to implement the classification rule in the ingress direction on the CE interface connected to customer LAN.

The figure below describes a sample service description of qos-classification for a site :

```
   <service>
       <qos>
           <qos-classification-policy>
               <rules>
                   <id>1</id>
                   <match-flow>
                       <ipv4-src-prefix>192.0.2.0/24</ipv4-src-prefix>
                       <ipv4-dst-prefix>203.0.113.1/32</ipv4-dst-
prefix>
                       <l4-dst-port>80</l4-dst-port>
                       <l4-protocol>tcp</l4-protocol>
                   </match-flow>
                   <target-class-id>DATA2</target-class-id>
               </rules>
               <rules>
                   <id>2</id>
                   <match-flow>
                       <ipv4-src-prefix>192.0.2.0/24</ipv4-src-prefix>
                       <ipv4-dst-prefix>203.0.113.1/32</ipv4-dst-
prefix>
                       <l4-dst-port>21</l4-dst-port>
                       <l4-protocol>tcp</l4-protocol>
                   </match-flow>
                   <target-class-id>DATA2</target-class-id>
               </rules>
               <rules>
                   <id>3</id>
                   <target-class-id>DATA1</target-class-id>
               </rules>
           </qos-classification-policy>
       </qos>
   </service>
```

In the example above :

- HTTP traffic from 192.0.2.0/24 LAN destinated to 203.0.113.1/32 will be classified in DATA2.

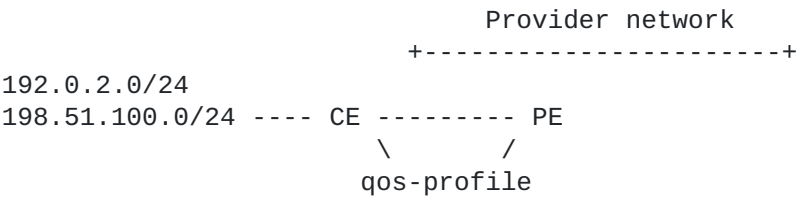
- FTP traffic from 192.0.2.0/24 LAN destinated to 203.0.113.1/32 will be classified in DATA2.

- All other traffic will be classified in DATA1.

The order of rules is really important. The management system responsible for translating those rules in network element configuration MUST keep the same processing order in element configuration. The order of rule is defined by the "id" leaf. The lowest "id" MUST be processed first.

5.2.6.1.2. QoS profile

User can choose between standard profile provided by the operator or custom profile.



A custom qos-profile is defined as a list of class of services and associated properties. The properties are :

- rate-limit : used to rate-limit the class of service. The value is expressed as a percentage of the global service bandwidth.

- priority-level : used to define priorities between class of services. The value of the priority to be used is dependant of each administration. The higher the priority-level is, the higher the priority of the class will be. Priority-level is used to define strict priority queueing. A priority-level 250 class will be served before a priority-level 100 class until there is no more packet to process or until rate-limit does not allow anymore packets from the higher priority class.

- guaranteed-bw-percent : used to define a guaranteed amount of bandwidth for the class of service. It is expressed as a percentage. The guaranteed-bw-percent uses available bandwidth at the priority-level of the class.

Example of service configuration using a standard qos profile :

```
   <site>
       <site-id>1245HRTFGJGJ154654</site-id>
       <service>
           <svc-input-bandwidth>100000000</svc-input-bandwidth>
           <svc-output-bandwidth>100000000</svc-output-bandwidth>
           <qos>
               <qos-profile>
                   <profile>PLATINUM</profile>
               </qos-profile>
           </qos>
       </service>
   </site>
   <site>
       <site-id>555555AAAA2344</site-id>
       <service>
           <svc-input-bandwidth>2000000</svc-input-bandwidth>
           <svc-output-bandwidth>2000000</svc-output-bandwidth>
           <qos>
               <qos-profile>
                   <profile>GOLD</profile>
               </qos-profile>
           </qos>
       </service>
   </site>
```

Example of service configuration using a custom qos profile :

```
 <site>
     <site-id>Site1</site-id>
     <service>
         <svc-input-bandwidth>100000000</svc-input-bandwidth>
         <svc-output-bandwidth>100000000</svc-output-bandwidth>
         <qos>
             <qos-profile>
                 <classes>
                     <class>
                         <class-id>REAL_TIME</class-id>
                         <rate-limit>10</rate-limit>
                         <priority-level>10</priority-level>
                     </class>
                     <class>
                         <class-id>DATA</class-id>
                         <priority-level>5</priority-level>
                     </class>
                 </classes>
             </qos-profile>
         </qos>
```

```
    </service>
 </site>
 <site>
    <site-id>Site2</site-id>
    <service>
        <svc-input-bandwidth>2000000</svc-input-bandwidth>
        <svc-output-bandwidth>2000000</svc-output-bandwidth>
        <qos>
            <qos-profile>
                <classes>
                    <class>
                      <class-id>REAL_TIME</class-id>
                      <rate-limit>30</rate-limit>
                      <priority-level>10</priority-level>
                    </class>
                    <class>
                      <class-id>DATA1</class-id>
                      <priority-level>5</priority-level>
                      <guaranteed-bw-percent>80</guaranteed-bw-
percent>
                    </class>
                    <class>
                      <class-id>DATA2</class-id>
                      <priority-level>5</priority-level>
                      <guaranteed-bw-percent>20</guaranteed-bw-
percent>
                    </class>
                </classes>
            </qos-profile>
        </qos>
    </service>
 </site>
```

The custom qos-profile for site1 defines that traffic from REAL_TIME class will have a higher priority than traffic from DATA class. The REAL_TIME traffic will be rate-limit to 10% of the service bandwidth (10% of 100Mbps = 10Mbps) to let some place for DATA traffic.

The custom qos-profile for site2 defines that traffic from REAL_TIME class will have a higher priority than traffic from data traffic. Data traffic will be splitted in two class of service DATA1 and DATA2 that will share bandwidth between them according to the percentage of guaranteed-bw-percent. The maximum of percentage to be used is not limited by this model but MUST be limited by the management system according to the policies authorized by the service provider. The REAL_TIME traffic will be rate-limit to 30% of the service bandwidth (30% of 100Mbps = 30Mbps) to let some place for data traffic. In case of congestion of the access, the REAL_TIME traffic can go up to 30Mbps (Let's assume that 20Mbps only are consumed). The DATA1 and

DATA2 will share remaining bandwidth (80Mbps) according to their percentage. So DATA1 will be served with at least 64Mbps of bandwidth.

5.2.6.2. Multicast

The multicast section defines the type of site in the customer multicast topology : source, receiver, or both. These parameters will help management system to optimize the multicast service. User can also define the type of multicast relation with the customer : router (requires a protocol like PIM), host (IGMP or MLD), or both. Transport protocol (IPv4 or IPv6 or both) can also be defined.

5.2.7. Site network accesses

As mentioned, a site may be multihomed. Each network access for a site is defined in the site-network-accesses list. The site-network-access defines how the service is connected on the network and is splitted in three main classes of parameters :

- bearer : defines physical parameters of the attachment.
- connection : defines protocol parameters of the attachment (transport layer and routing protocols).
- availability : defines the site availability policy. Availability is defined in Section 5.2.8.3

Some parameters from the site can be configured also at the access level like : routing, services, security ... Defining parameters only at site level will provide inheritance. If a parameter is configured at both site and access level, the access level parameter MUST override the site level parameter.

5.2.7.1. Bearer

Bearer defines an internal reference to the bearer used for the access. Two strings are available (type and bearer-reference) to encode necessary informations to map the VPN access to the appropriate network access bearer.

How the mapping is done is out of scope of the document.

5.2.7.2. Connection

The connection defines the protocol parameters of the attachment (IPv4 and IPv6). Depending of the management mode, it refers to the PE-CE addressing or CE to customer LAN addressing. In any case, it

describes the provider to customer responsibility boundary. For a customer managed site, it refers to the PE-CE connection. For a provider managed site, it refers to the CE to LAN connection.

5.2.7.2.1. IP addressing

IP subnet can be configured for either transport protocols. For a dual stack connection, two subnets will be provided, one for each transport layer.

The address-allocation-type will help in defining how the address allocation MUST be done. The current model proposes three ways of IP address allocation :

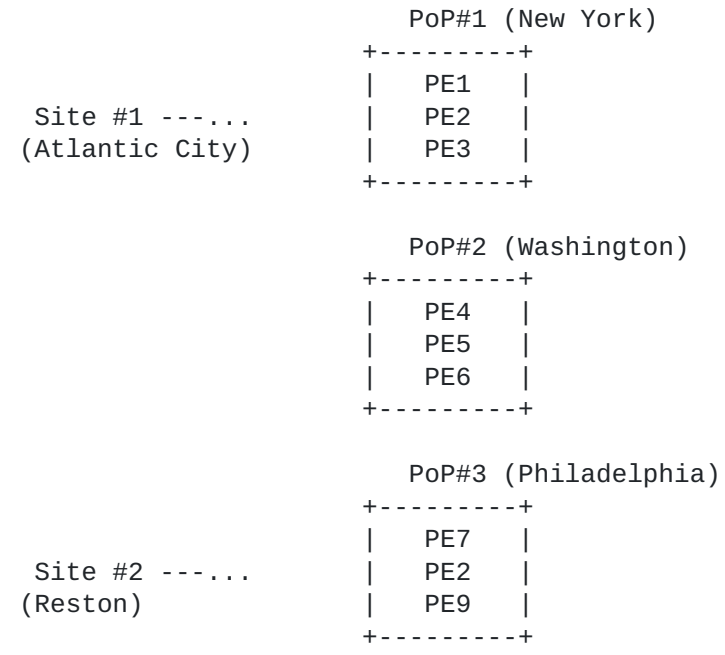
- provider-dhcp : the provider will provide DHCP service for customer equipments, this is applicable to both IPv4 and IPv6 addressing.
- static-address : Addresses will be assigned manually on both sides, this is applicable to both IPv4 and IPv6 addressing.
- slaac : enables stateless address autoconfiguration ([RFC4862]). This is applicable only for IPv6.

5.2.8. Deciding where to connect the site

The management system will have to decide where to connect the site in the provider network (PE, aggregation switch ...). This decision MAY be based on any constraint that are up to the service provider : least load, distance ... The current model proposes some parameters that will help the management system to decide where to attach the customer site. It would be up to the service provider to define which on those parameters are relevant for placing the site, moreover the service provider can decide to rely also on other internal parameters.

5.2.8.1. Site location

The location information provided in this model MAY be used by a management system to decide the target PE to mesh the site.



In the example above, the management system may decide to mesh Site #1 on a PE from Philadelphia PoP for distance reason. It may also take into account resources available on PEs to decide the exact target PE (least load). In case of shortest distance PE used, it may also decide to mesh Site #2 on Washington PoP.

5.2.8.2. Site diversity

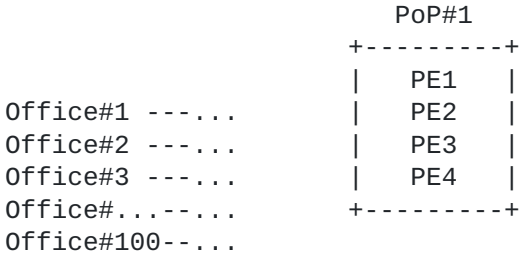
The site diversity defines what is the acceptable fate sharing level in case multiple sites for a single VPN must be provisioned in a common location. The site diversity introduces the notion of site-group. Sites belonging to the same site-group cannot share the same fate. We propose to introduce two constraints :

PoP diverse : site belonging to the same site-group MUST be provisioned on different PoPs.

PE diverse : site belonging to the same site-group MUST be provisioned on different PE routers.

How these diversity constraints are applied is out of scope of the document. As an example, the management system receiving the request for diversity, MAY exchange information with some OSS components to define the best target PEs based on location and resource availability.

As example, if a company has multiple small branch offices (single homed) that requires to be connected in the same location, it is desirable to dispatch the attachment on multiple PEs. So in case of PE crash, only some offices will be impacted.

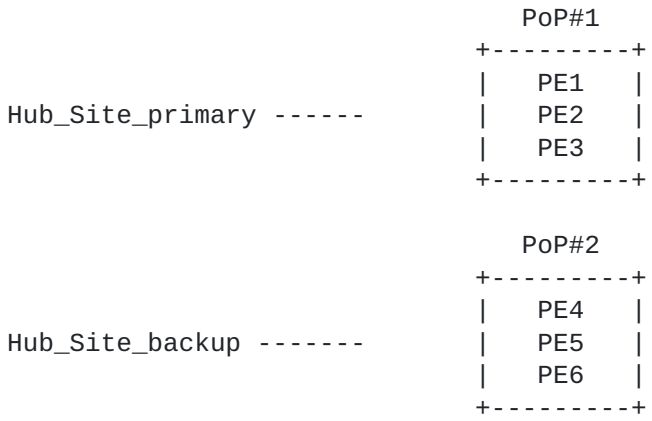


In the figure above, it may be good to mesh 25 offices on each PE of PoP#1 to prevent concentration of two many customer offices on common network elements.

5.2.8.3. Site network access availability

A site may be multihomed, so having multiple site-network-accesses. An implementation MAY apply placement diversity for accesses belonging to the same site. By default, diversity for accesses belonging to the same site is set to "PE-diverse".

Consider a dual homed hub site, it is desirable for redundancy to provision the two VPN access connections on two different PEs or two different PoPs.

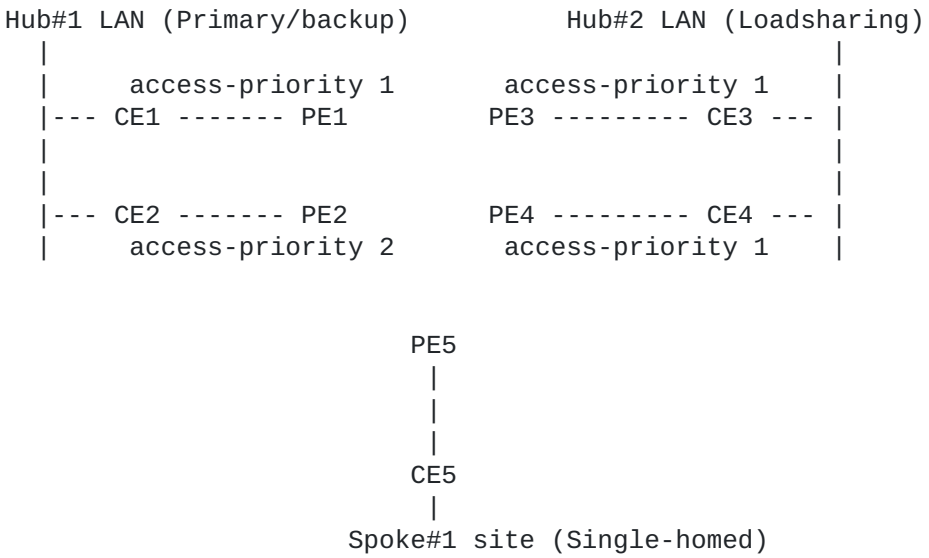


In a PoP diverse scenario, the management system may decide to mesh Hub_Site_primary on any PE of PoP#1 and Hub_Site_backup on any PE of PoP#2. In a PE diverse scenario, if the management system decides to

mesh Hub_Site_primary on PE1, it is require to mesh Hub_Site_backup on any PE different from PE1.

The site-network-access/availability defines parameters for the site redundancy. The access-priority defines a preference for a particular access. This preference is used to model any kind of loadbalancing or primary/backup scenario. The highest the access-priority is, and the highest the preference will be.

The figure below describes how access-priority attribute can be used.

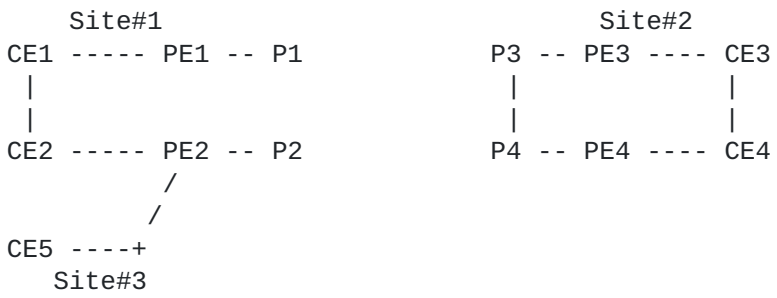


In the figure above, Hub#2 requires loadsharing so all the site-network-accesses must use the same access-priority value. At the contrary, as Hub#1 requires primary/backup, a higher access-priority will be configured on the primary access.

More complex scenario can be modeled. Let's consider a Hub site with 5 accesses to the network (A1,A2,A3,A4,A5). The customer wants to loadshare traffic on A1,A2 in the nominal situation. If A1 and A2 fails, he wants to loadshare traffic on A3 and A4, and finally if A1 to A4 are down, he wants to use A5. We can model it easily by associating the following access-priorities : A1=100, A2=100, A3=50, A4=50, A5=10.

5.2.8.3.1. Traffic protection

The service model supports the ability to protect traffic for the site. Protection provides a better availability to multihoming by using local-repair techniques in case of failures.



In the figure above, we consider an IPVPN service with three sites including two dual homed sites (site#1 and #2). For dual homed sites, we consider PE1-CE1 and PE3-CE3 as primary, and PE2-CE2,PE4-CE4 as backup for the example (even if protection also applies to loadsharing scenarios.)

In order to protect Site#2 against a failure, user may set the enabled leaf of traffic-protection to true on the site-network-accesses of site#2. How the traffic protection will be implemented is out of scope of the document. But as an example, in such case, if we consider traffic coming from a remote site (site#1 or site#3), primary path is to use PE3 as egress PE. PE3 may have preprogrammed a backup forwarding entry pointing to backup path (through PE4-CE4) for all prefixes going through PE3-CE3 link. How backup path is computed is out of scope of the document. When PE3-CE3 link fails, traffic is still received by PE3 but PE3 switch automatically traffic to the backup entry, path will so be PE1-P1-(...)-P3-PE3-PE4-CE4 until remote PEs reconverge and use PE4 as egress PE.

5.2.8.4. Route Distinguisher and VRF allocation

Route distinguisher is also a critical parameter of PE-based L3VPN as described in [RFC4364] that will allow to distinguish common addressing plans in different VPNs. As for Route-targets, it is expected management system to allocate a VRF on the target PE and a route-distinguisher for this VRF.

If a VRF exists on the target PE, and the VRF fulfils the connectivity constraints for the site, there is no need to recreate another VRF and the site MAY be meshed within this existing VRF. How the management system checks that an existing VRF fulfils the connectivity constraints for a site is out of scope of this document.

If no VRF exists on the target PE, filling the site constraints, the management system will have to initiate a new VRF creation on the target PE and will have to allocate a new route distinguisher for this new VRF.

The management system MAY apply a per-VPN or per-VRF allocation policy for the route-distinguisher depending of the service provider policy. In a per-VPN allocation policy, all VRFs (dispatched on multiple PEs) within a VPN will share the same route distinguisher value. In a per-VRF model, all VRFs will always have a unique route-distinguisher value. Some other allocation policies are also possible, and this document does not restrict the allocation policies to be used.

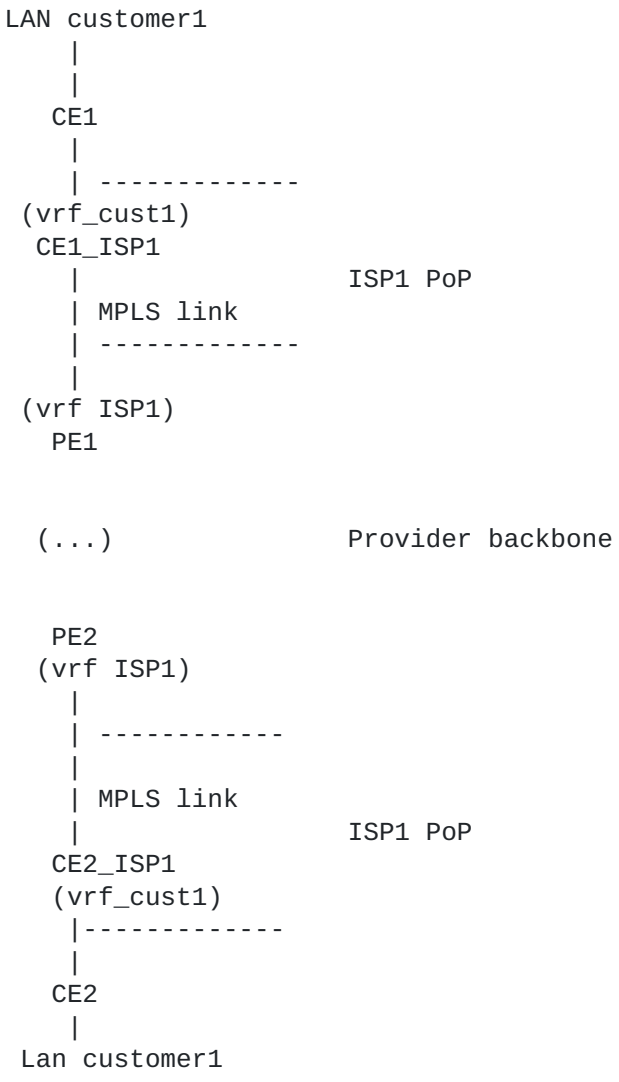
Allocation of route-distinguisher MAY be done in the same way as the route-targets. The example provided in Section 5.1.1.1 could be reused.

Note that a service provider MAY decide to configure target PE for automated allocation of route distinguisher. In this case, there will be no need for any backend system to allocate a route-distinguisher value.

5.3. Enhanced VPN features

5.3.1. Carrier Supporting Carrier

In case of Carrier Supporting Carrier (CsC), a customer MAY want to build MPLS service using an IPVPN as transport layer.



In the figure above, ISP1 resells IPVPN service but has no transport infrastructure between its PoPs. ISP1 uses an IPVPN as transport infrastructure (belonging to another provider) between its PoPs.

In order to support CsC, the VPN service must be declared MPLS support using the "mpls" leaf set to true in vpn-svc. The link between CE1_ISP1/PE1 and CE2_ISP1/PE2 must also run a MPLS signalling protocol. This configuration is done at the site level.

In the proposed model, LDP or BGP can be used as MPLS signalling protocol. In case of LDP, an IGP routing protocol MUST also be

activated. In case of BGP signalling, BGP MUST also be configured as routing-protocol.

5.3.2. Transport constraints

A customer may require some constraints for transporting traffic between particular sites. As example, a customer may require low latencies and disjoint paths between two hub sites. The current model proposes to define a list of constraints that can be augmented for unicast and/or multicast traffic. For unicast traffic, the model considers that the constraints are bidirectional (same constraint from site1 to site2 and site2 to site1). For multicast, constraints are unidirectional from source to receiver. The current model supports the following constraints :

- Latency : this constraint allow to create the lowest latency path possible or to create a path with a latency boundary. In case a latency boundary is required, the boundary MUST be encoded in the constraint-opaque-value using a millisecond unit.

- Bandwidth : this constraint allow to create a path that fits specific bandwidth requirement. If no constraint-opaque-value is provided, an implementation SHOULD use the lowest bandwidth between the two sites as reference. If constraint-opaque-value is used, the required bandwidth MUST be encoded in Mbps, and the implementation MUST use this value as reference.

- Jitter : this constraint allow to create a path with a jitter boundary. constraint-opaque-value MUST be used with jitter constraint and MUST contain the jitter boundary expressed in milliseconds.

- Path diversity : this constraint allow creation of disjoint paths between two sites. This requires the sites to be multihomed. constraint-opaque-value is not used.

- Site diversity : this constraint is similar to path diversity but ensures that paths are not crossing the same sites. This requires the sites to be multihomed. constraint-opaque-value MAY be used to encode additional site location that must be avoided.

5.4. Using configuration templates

The proposed model supports the creation and application of configuration templates for sites.

A template can be configured by creating adding a site in the site-template list. The "site-templates" list contains only templates. Real sites are part of the "sites" list.

Multiple templates can be configured. Templates can be applied at multiple levels referenced by apply-template leaf. The apply-template references the site-id of the template to be called. The location of the apply-template within the sites hierarchy defines which parameters must be inherited. For example, if apply-template is done on service container of a site, only service container parameters (and childs) from the template will be applied.

```
   <site-template>
    <site-id>Template-VoiceCoS-Cust1</site-id>
     <service>
      <qos>
       <qos-profile>
        <classes>
         <class>
          <class-id>REAL_TIME</class-id>
          <rate-limit>30</rate-limit>
          <priority-level>10</priority-level>
         </class>
         <class>
          <class-id>DATA1</class-id>
          <priority-level>5</priority-level>
          <guaranteed-bw-percent>80</guaranteed-bw-percent>
         </class>
         <class>
          <class-id>DATA2</class-id>
          <priority-level>5</priority-level>
          <guaranteed-bw-percent>20</guaranteed-bw-percent>
         </class>
        </classes>
      </qos-profile>
     </qos>
    </service>
   </site-template>


   <site-template>
       <site-id>Template-VPNsite-Customer1</site-id>
       <service>
           <qos>
               <qos-profile>
                   <profile>PLATINUM</profile>
               </qos-profile>
           </qos>
       </service>
   </site-template>
```

Site-templates allow to define configuration blocks that will be inherited by one or multiple sites in order to speed up configuration. For example, if all the sites of an IPVPN service have the almost same configuration (routing-protocol, qos, management ...), a template can be created and each site of the VPN will reference the template. If a site has some particular parameters, specific parameters within the site MUST always override parameters derived from template.

The example above defines two site templates :

- Template-VPNsite-Customer1 that will be used to configure all the VPN sites for customer 1.

- Template-VoiceCoS-Cust1 that will be used to configure some special CoS policy on some specific accesses of the VPN.

In the example below, all sites of VPN1 are inheriting basic configuration from template Template-VPNsite-Customer1. Some specific parameters like svc-input-bandwidth are also defined for each site. For Site 4 and 5 , specific QoS parameters are required, a new template Template-VoiceCoS-Cust1 is applied at service level for these two sites, overriding the service parameters from the Template-VPNsite-Customer1 template.

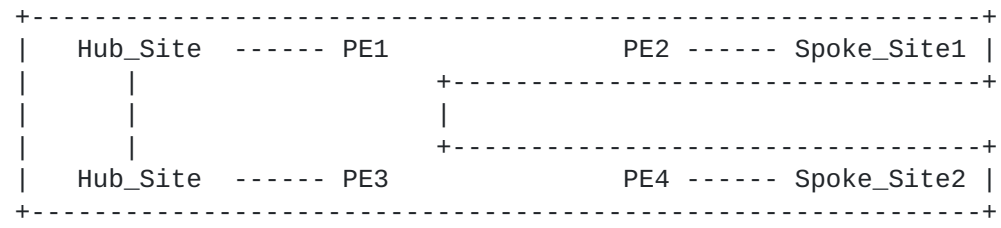
```
   <site>
       <site-id>Site1</site-id>
       <apply-template>Template-VPNsite-Customer1</apply-template>
       <service>
           <svc-input-bandwidth>5000000</svc-input-bandwidth>
           <svc-output-bandwidth>5000000</svc-output-bandwidth>
       </service>
   </site>
   <site>
       <site-id>Site2</site-id>
       <apply-template>Template-VPNsite-Customer1</apply-template>
       <service>
           <svc-input-bandwidth>20000000</svc-input-bandwidth>
           <svc-output-bandwidth>20000000</svc-output-bandwidth>
       </service>
   </site>
   <site>
       <site-id>Site3</site-id>
       <apply-template>Template-VPNsite-Customer1</apply-template>
       <service>
           <svc-input-bandwidth>30000000</svc-input-bandwidth>
           <svc-output-bandwidth>30000000</svc-output-bandwidth>
       </service>
   </site>


   <site>
       <site-id>Site4</site-id>
       <apply-template>Template-VPNsite-Customer1</apply-template>
       <service>
           <apply-template>Template-VoiceCoS-Cust1</apply-template>
           <svc-input-bandwidth>100000000</svc-input-bandwidth>
           <svc-output-bandwidth>100000000</svc-output-bandwidth>
       </service>
   </site>
   <site>
       <site-id>Site5</site-id>
       <apply-template>Template-VPNsite-Customer1</apply-template>
       <service>
           <apply-template>Template-VoiceCoS-Cust1</apply-template>
           <svc-input-bandwidth>450000000</svc-input-bandwidth>
           <svc-output-bandwidth>450000000</svc-output-bandwidth>
       </service>
   </site>
```

6. Service model usage example

As explained in Section 4, this service model is intended to be instantiated at a management layer and is not intended to be used directly on network elements. The management system serves as a central point of configuration of the overall service.

This section provides an example on how a management system can use this model to configure an IPVPN service on network elements.

The example wants to achieve the provisionning of a VPN service for 3 sites using hub and spoke topology. One of the site will be dual homed and loadsharing is expected.



The following XML describes the overall simplified service configuration of this VPN.

```
<vpn-svc>
    <vpn-id>12456487</vpn-id>
    <customer-name>CUSTOMER1</customer-name>
    <topology>hub-spoke</topology>
</vpn-svc>
```

When receiving the request for provisioning the VPN service, the management system will internally (or through discussion with other OSS component) allocates VPN route-targets. In this specific case two RTs will be allocated (100:1 for Hub and 100:2 for Spoke). The output below describes the configuration of Spoke1.

```
<site>
    <site-id>Spoke_Site1</site-id>
    <site-diversity>
        <type>pe-diverse</type>
        <site-group>100</site-group>
    </site-diversity>
    <location>
        <city-code>NY</city-code>
        <country-code>US</country-code>
    </location>
    <routing-protocols>
      <routing-protocol>
        <type>bgp</type>
        <bgp>
            <autonomous-system>500</autonomous-system>
            <address-family>ipv4-unicast</address-family>
        </bgp>
      </routing-protocol>
    </routing-protocols>
    <site-network-accesses>
        <site-network-access>
            <site-network-access-id>Spoke_Site1</site-network-access-
id>
            <ip-connection>
                <ipv4>
                    <provider-address>203.0.113.254</provider-address>
                    <customer-address>203.0.113.2</customer-address>
                    <mask>24</mask>
                </ipv4>
            </ip-connection>
            <vpn-attachment>
                <vpn-id>12456487</vpn-id>
                <site-role>spoke-role</site-role>
            </vpn-attachment>
        </site-network-access>
    </site-network-accesses>
    <management>
        <type>provider-managed</type>
        <management-transport>ipv4-unicast</management-transport>
        <address>192.0.2.1</address>
    </management>
    <service>
        <svc-input-bandwidth>450000000</svc-input-bandwidth>
        <svc-output-bandwidth>450000000</svc-output-bandwidth>
    </service>


</site>
```

When receiving the request for provisioning Spoke1 site, the management system MUST allocate network resources for this site. It MUST first decide the target network elements to provision the access, and especially the PE router (and may be an aggregation switch). As described in Section 5.2.8, the management system SHOULD use the location information and SHOULD use the site-diversity constraint to find the appropriate PE. In this case, we consider Spoke1 requires PE diversity with Hub and that management system allocate PEs based on lowest distance. Based on the location information, the management system finds the available PEs in the nearest area of the customer and picks one that fits the site-diversity constraint.

When the PE is chosen, management system needs to allocate interface resources on the node, one interface is so picked from the PE available pool. The management system can start provisioning the PE node by using any mean (Netconf, CLI, ...). The management system will check if a VRF is already present that fits the needs. If not, it will provision the VRF : Route distinguisher will come from internal allocation policy model, route-targets are coming from the vpn-policy configuration of the site (management system allocated some RTs for the VPN). As the site is a spoke site (site-role), the management system knows which RT must be imported and exported. As the site is provider managed, some management route-targets may also be added (100:5000). Standard provider VPN policies MAY also be added in the configuration.

Example of generated PE configuration :

```
ip vrf Customer1
 export-map STD-CUSTOMER-EXPORT      <---- Standard SP configuration
 route-distinguisher 100:3123234324
 route-target import 100:1
 route-target import 100:5000        <---- Standard SP configuration
 route-target export 100:2                   for provider managed
!
```

When the VRF has been provisioned, the management system can start configuring the access on the PE using the allocated interface information. IP addressing is derived from the subnet-prefix of the connection. One address will be picked from the subnet for the PE, another will be used for the CE configuration. Routing protocols will also be configured on the PE, bgp will be used as requested in the service model. Peering addresses will be derived from subnet-prefix. PE AS number is well known and as CE is provider managed, CE AS number can be automatically allocated by the management system. Some provider standard configuration templates may also be added.

Example of generated PE configuration :

```
interface Ethernet1/1/0.10
 encapsulation dot1q 10
 ip vrf forwarding Customer1
 ip address 198.51.100.1 255.255.255.252 <---- Comes from
                                                automated
allocation
 ip access-group STD-PROTECT-IN     <---- Standard SP config
!
router bgp 100
 address-family ipv4 vrf Customer1
  neighbor 198.51.100.1 remote-as 65000   <---- Comes from
                                              automated allocation
  neighbor 198.51.100.1 route-map STD in  <---- Standard SP config
  neighbor 198.51.100.1 filter-list 10 in <---- Standard SP config
!
ip route vrf Customer1 192.0.2.1 255.255.255.255 198.51.100.2
! Static route for provider administration of CE
!
```

As the CE router is not reachable at this stage, the management system can produce a complete CE configuration that can be uploaded to the node by manual operation before sending the CE to customer premise. The CE configuration will be built as for the PE. Based on the CE type (vendor/model) allocated to the customer and bearer information, the management system knows which interface must be configured on the CE. PE-CE link configuration is expected to be handled automatically using the service provider OSS as both resources are managed internally. CE to LAN interface parameters like IP addressing are derived from ip-connection taking into account how management system distributes addresses between PE and CE within the subnet. This will allow to produce a plug'n'play configuration for the CE.

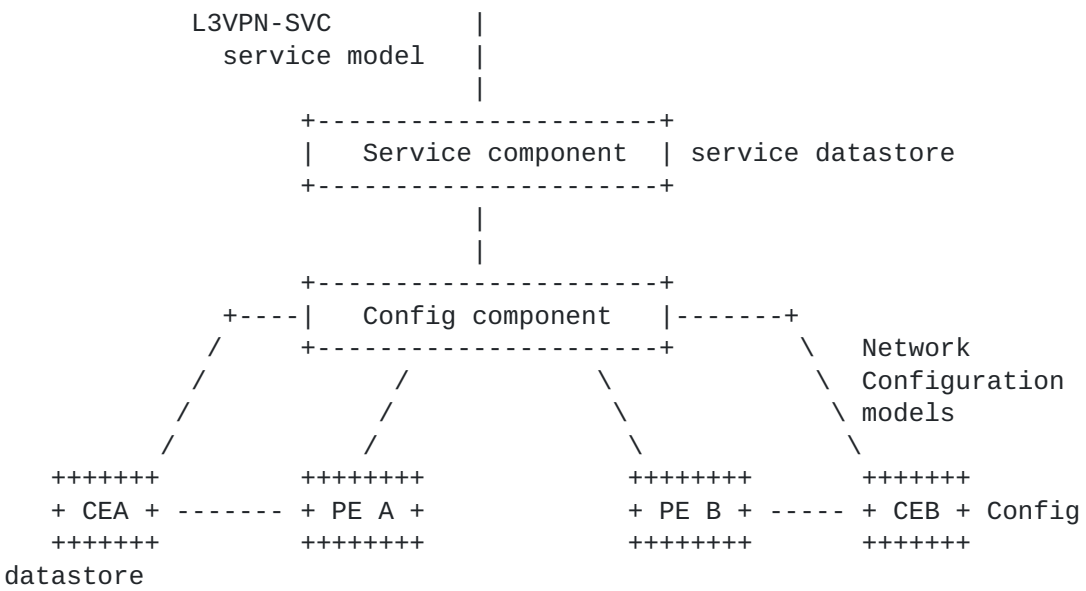
Example of generated CE configuration :

```
interface Loopback10
 description "Administration"
 ip address 192.0.2.1 255.255.255.255
!
interface FastEthernet10
 description "WAN"
 ip address 198.51.100.2 255.255.255.252 <---- Comes from
                                              automated allocation
!
interface FastEthernet11
 description "LAN"
 ip address 203.0.113.254 255.255.255.0 <---- Comes from
                                         ip-connection
!
router bgp 65000
  redistribute static route-map STATIC2BGP <---- Standard SP
                                                 configuration
  neighbor 198.51.100.1 remote-as 100     <---- Comes from
                                            automated allocation
  neighbor 203.0.113.2 remote-as 500     <---- Comes from
                                            ip-connection
!
route-map STATIC2BGP permit 10
 match tag 10
!
```

7. Interaction with Other YANG Modules

As expressed in Section 4, this service module is intended to be instantiated in management system and not directly on network elements.

It will be the role of the management system to configure the network elements. The management system MAY be modular, so the component instantiating the service model (let's call it service component) and the component responsible for network element configuration (let's call it configuration component) MAY be different.



In the previous sections, we provided some example of translation of service provisioning request to router configuration lines as illustration. In the NetConf/Yang ecosystem, it will be expected NetConf/YANG to be used between configuration component and network elements to configure the requested service on these elements.

In this framework, it is expected from standardization to also work on specific configuration YANG modelization of service components on network elements. There will be so a strong relation between the abstracted view provided by this service model and the detailed configuration view that will be provided by specific configuration models for network elements.

Authors of this document are expecting definition of YANG models for network elements on this non exhaustive list of items :

- VRF definition including VPN policy expression.
- Physical interface.
- IP layer (IPv4, IPv6).
- QoS : classification, profiles...
- Routing protocols : support of configuration of all protocols listed in the document, as well as routing policies associated with these protocols.

- Multicast VPN.
- Network Address Translation.
- ...

Example of VPN site request at service level using this model :

```
  <site>
   <site-id>Site A</site-id>
   <site-network-accesses>
    <site-network-access>
     <ip-connection>
      <ipv4>
       <address-allocation-type>static-address</address-allocation-
type>
       <subnet-prefix>203.0.113.0/30</subnet-prefix>
      </ipv4>
     </ip-connection>
     <vpn-attachment>
      <vpn-policy-id>VPNPOL1</vpn-policy-id>
     </vpn-attachment>
    </site-network-access>
   </site-network-accesses>
   <routing-protocols>
    <routing-protocol>
     <type>static</type>
     <static>
      <cascaded-lan-prefixes>
       <ipv4-lan-prefixes>
        <lan>198.51.100.0/30</lan>
        <next-hop>203.0.113.2</next-hop>
       </ipv4-lan-prefixes>
      </cascaded-lan-prefixes>
     </static>
    </routing-protocol>
   </routing-protocols>
   <management>
    <type>customer-managed>/type<
   </management>
   <vpn-policy-list>
    <vpn-policy>
      <vpn-policy-id>VPNPOL1</vpn-policy-id>
      <entries>
          <id>1</id>
          <vpn>
              <vpn-id>VPN1</vpn-id>
              <site-role>any-to-any-role</site-role>
          </vpn>
      </entries>
    </vpn-policy>
   </vpn-policy-list>
  </site>
```

In the service example above, it is expected that the service component requests to the configuration component of the management system the configuration of the service elements. If we consider

that service component selected a PE (PE A) as target PE for the site, the configuration component will need to push the configuration to PE A. The configuration component will use several YANG data models to define the configuration to be applied to PE A. The XML configuration of PE-A may look like this :

```
<if:interfaces>
     <if:interface>
      <if:name>eth0</if:name>
      <if:type>ianaift:ethernetCsmacd</if:type>
      <if:description>
       Link to CEA.
      </if:description>
      <ip:ipv4>
       <ip:address>
        <ip:ip>203.0.113.1</ip:ip>
        <ip:prefix-length>30</ip:prefix-length>
       </ip:address>
       <ip:forwarding>true</ip:forwarding>
      </ip:ipv4>
     </if:interface>
</if:interfaces>
<rt:routing>
     <rt:routing-instance>
      <rt:name>VRF_CustA</rt:name>
      <rt:type>l3vpn:vrf</rt:type>
      <rt:description>VRF for CustomerA</rt:description>
      <l3vpn:route-distinguisher>
      100:1546542343
      </l3vpn:route-distinguisher>
      <l3vpn:import-rt>100:1</l3vpn:import-rt>
      <l3vpn:export-rt>100:1</l3vpn:export-rt>
      <rt:interfaces>
       <rt:interface>
        <rt:name>eth0</rt:name>
       </rt:interface>
      </rt:interfaces>
      <rt:routing-protocols>
       <rt:routing-protocol>
        <rt:type>rt:static</rt:type>
        <rt:name>st0</rt:name>
        <rt:static-routes>
         <v4ur:ipv4>
          <v4ur:route>
           <v4ur:destination-prefix>
           198.51.100.0/30
           </v4ur:destination-prefix>
           <v4ur:next-hop>
```

```
              <v4ur:next-hop-address>
              203.0.113.2
              </v4ur:next-hop-address>
             </v4ur:next-hop>
            </v4ur:route>
           </v4ur:ipv4>
          </rt:static-routes>
         </rt:routing-protocol>
        </rt:routing-protocols>
       </rt:routing-instance>
   </rt:routing>
```

8. YANG Module

```
<CODE BEGINS> file "ietf-l3vpn-svc@2016-03-11.yang"

module ietf-l3vpn-svc {
    namespace "urn:ietf:params:xml:ns:yang:ietf-l3vpn-svc";

    prefix l3vpn-svc;

    import ietf-routing {
            prefix "rt";
        }

    import ietf-inet-types {
        prefix inet;
    }

    import ietf-yang-types {
        prefix yang;
    }

    organization
     "IETF L3SM Working Group";

    contact
        "WG List:   &lt;mailto:l3sm@ietf.org&gt;

        Editor:

        ";

    description
        "The YANG module defines a generic service configuration
        model for Layer 3 VPN common across all of the vendor
        implementations.";
```

```
    revision 2016-03-11 {
        description
        "
            * Modify VPN policy and creating a vpn-policy-list
            * Add VPN policy reference and VPN ID reference
            under site-network-access
        ";
        reference "draft-ietf-l3sm-l3vpn-service-yang-05";
    }
    revision 2016-01-04 {
        description
        "
            * Add extranet-vpn container in vpn-svc
            * Creating top level containers
            * Refine groupings
            * Added site-vpn-flavor
        ";
        reference "draft-ietf-l3sm-l3vpn-service-yang-03";
    }
    revision 2016-01-04 {
        description
         "
            * qos-profile moved to choice
            * vpn leaf moved to vpn-id in vpn-policy
            * added ordered-by user to qos classification list
            * moved traffic protection to access availability
            * creating a choice in matching filter for VPN policy
            * added dot1p matching field in flow-definition
        ";
        reference "";

    }
    revision 2015-12-07 {
        description
         "
            * A site is now a collection of site-accesses.
            This was introduced to support M to N availability.
            * Site-availability has been removed, replaced by
            availability parameters under site-accesses
            * Added transport-constraints within vpn-svc
        ";
        reference "draft-ietf-l3sm-l3vpn-service-yang-02";

    }

    revision 2015-11-03 {
        description "
        * Add ToS support in match-flow
```

```
        * nexthop in cascaded lan as mandatory
        * customer-specific-info deleted and moved to routing
        protocols
        * customer-lan-connection modified : need prefix and CE address
        * add choice in managing PE-CE addressing
        * Simplifying traffic protection
        ";
        reference "";
    }
    revision 2015-09-10 {
        description "
        * Refine groupings for vpn-svc
        * Removed name in vpn-svc
        * id in vpn-svc moved to string
        * Rename id in vpn-svc to vpn-id
        * Changed key of vpn-svc list to vpn-id
        * Add DSCP support in flow definition
        ";
        reference "";
    }
    revision 2015-08-07 {
        description
         "
          Multicast :
            * Removed ACL from security
            * Add FW for site and cloud access
         ";
        reference "";
    }
    revision 2015-08-05 {
        description
         "
          Multicast :
          * Removed anycast-rp identity as discovery mechanism
          * Added rp-group mappings for multicast
          * Added flag for provider managed RP.
         ";
        reference "";
    }
    revision 2015-08-03 {
        description
         " * Creating multiple reusable groupings
           * Added mpls leaf in vpn-svc for carrier's carrier case
           * Modify identity single to single-site
           * Modify site-type to site-role and also child identities.
           * Creating OAM container under site and moved BFD in.
           * Creating flow-definition grouping to be reused
           in ACL, QoS ...
```

```
           * Simplified VPN policy.
           * Adding multicast static group to RP mappings.
           * Removed native-vpn and site-role from global site
           cfg, now managed within the VPN policy.
           * Creating a separate list for site templates.
         ";
        reference "draft-ietf-l3sm-l3vpn-service-yang-01";
    }
    revision 2015-07-02 {
        reference "draft-ietf-l3sm-l3vpn-service-yang-00";
    }
    revision 2015-04-24 {
        description "
        * Add encryption parameters
        * Adding holdtime for BFD.
        * Add postal address in location
        ";
        reference "draft-lstd-l3sm-l3vpn-service-yang-00";
    }
    revision 2015-02-05 {
        description "Initial revision.";
        reference "draft-l3vpn-service-yang-00";
    }

    /* Typedefs */

    typedef svc-id {
        type string;
        description
         "Defining a type of service component
         identificators.";
    }

    typedef template-id {
        type string;
        description
         "Defining a type of service template
         identificators.";
    }

    typedef placement-diversity {
        type enumeration {
            enum "pop-diverse" {
                description
                 "The access must use another PoP
                 compared
                 to other accesses in the same group.";
            }
```

```
            enum "pe-diverse" {
                description
                 "The access must use another PE
                 compared
                 to other accesses in the same group.";
            }
        }
        description
         "Defines a type for service placement diversity.";
    }

    /* Identities */

    identity site-vpn-flavor {
        description
        "Base identity for the site VPN service flavor.";
    }
    identity site-vpn-flavor-single {
        base site-vpn-flavor;
        description
        "Base identity for the site VPN service flavor.
        Used when the site belongs to only one VPN.";
    }
    identity site-vpn-flavor-multi {
        base site-vpn-flavor;
        description
        "Base identity for the site VPN service flavor.
        Used when a logical connection of a site
        belongs to multiple VPNs.";
    }

    identity site-vpn-flavor-sub {
        base site-vpn-flavor;
        description
        "Base identity for the site VPN service flavor.
        Used when a site has multiple logical connections.
        Each of the connection may belong to different
        multiple VPNs.";
    }
/*
    identity site-vpn-flavor-sub {
        base site-vpn-flavor;
        description
        "Base identity for the site VPN service flavor.
        Used when a site has multiple logical connections and
        each logical connection belongs to a single
        particular VPN.";
    }
```

```
    identity site-vpn-flavor-submulti {
        base site-vpn-flavor;
        description
        "Base identity for the site VPN service flavor.
        Used when a site has multiple logical connections and
        each logical connection may belong to multiple VPNs with
        independent rules.";
    }
    */

    identity transport-constraint {
        description
         "Base identity for transport constraint.";
    }
    identity tc-latency {
        base transport-constraint;
        description
         "Base identity for transport constraint
         based on latency.";
    }
    identity tc-jitter {
        base transport-constraint;
        description
         "Base identity for transport constraint
         based on jitter.";
    }
    identity tc-bandwidth {
        base transport-constraint;
        description
         "Base identity for transport constraint
         based on bandwidth.";
    }
    identity tc-path-diversity {
        base transport-constraint;
        description
         "Base identity for transport constraint
         based on path diversity.";
    }
    identity tc-site-diversity {
        base transport-constraint;
        description
         "Base identity for transport constraint
         based on site diversity.";
    }

    identity management {
        description
         "Base identity for site management scheme.";
```

```
    }
    identity comanaged {
        base management;
        description
         "Base identity for comanaged site.";
    }
    identity customer-managed {
        base management;
        description
         "Base identity for customer managed site.";
    }
    identity provider-managed {
        base management;
        description
         "Base identity for provider managed site.";
    }

    identity address-allocation-type {
        description
         "Base identity for address-allocation-type
         for PE-CE link.";
    }
    identity pe-dhcp {
        base address-allocation-type;
        description
         "PE router provides DHCP service to CE.";
    }
    identity static-address {
        base address-allocation-type;
        description
         "PE-CE addressing is static.";
    }
    identity slaac {
        base address-allocation-type;
        description
         "Use IPv6 SLAAC.";
    }

/*
    identity site-availability {
        description
         "Base identity for site availability.";
    }
    identity loadsharing {
            base site-availability;
            description
             "Identity for loadsharing site.";
    }
```

```
    identity loadsharing-ibgp {
            base loadsharing;
            description
             "Identity for ECMP ibgp based loadsharing site.";
    }
    identity loadsharing-eibgp {
            base loadsharing;
            description
             "Identity for ECMP eibgp based loadsharing site.";
    }
    identity primary-backup {
            base site-availability;
            description
             "Identity for primary-backup site.";
    }
    identity single-site {
            base site-availability;
            description
             "Identity for single site.";
    }
    identity access-availability-type {
        description
         "base identity for access-availability-type";
    }
    identity primary-access {
        base access-availability-type;
        description
         "Identity for primary access type.";
    }
    identity backup-access {
        base access-availability-type;
        description
         "Identity for backup access type.";
    }
    identity single-access {
        base access-availability-type;
        description
         "Identity for single access type.";
    }
    identity loadsharing-access {
        base access-availability-type;
        description
         "Identity for loadsharing access type.";
    }
    */


    identity site-role {
```

```
        description
         "Base identity for site type.";
    }
    identity any-to-any-role {
        base site-role;
        description
         "Site in a any to any IPVPN.";
    }
    identity spoke-role {
        base site-role;
        description
         "Spoke Site in a Hub & Spoke IPVPN.";
    }
    identity hub-role {
        base site-role;
        description
         "Hub Site in a Hub & Spoke IPVPN.";
    }


    identity vpn-topology {
        description
         "Base identity for VPN topology.";
    }
    identity any-to-any {
        base vpn-topology;
        description
         "Identity for any to any VPN topology.";
    }
    identity hub-spoke {
        base vpn-topology;
        description
         "Identity for Hub'n'Spoke VPN topology.";
    }
    identity hub-spoke-disjoint {
        base vpn-topology;
        description
         "Identity for Hub'n'Spoke VPN topology
          where Hubs cannot talk between each other.";
    }

    identity multicast-tree-type {
        description
         "Base identity for multicast tree type.";
    }

    identity ssm-tree-type {
        base multicast-tree-type;
```

```
        description
         "Identity for SSM tree type.";
    }
    identity asm-tree-type {
        base multicast-tree-type;
        description
         "Identity for ASM tree type.";
    }
    identity bidir-tree-type {
        base multicast-tree-type;
        description
         "Identity for BiDir tree type.";
    }

    identity multicast-rp-discovery-type {
        description
         "Base identity for rp discovery type.";
    }
    identity auto-rp {
        base multicast-rp-discovery-type;
        description
         "Base identity for auto-rp discovery type.";
    }
    identity static-rp {
        base multicast-rp-discovery-type;
        description
         "Base identity for static type.";
    }
    identity bsr-rp {
        base multicast-rp-discovery-type;
        description
         "Base identity for BDR discovery type.";
    }

    identity routing-protocol-type {
        description
         "Base identity for routing-protocol type.";
    }

    identity ospf {
        base routing-protocol-type;
        description
         "Identity for OSPF protocol type.";
    }

    identity bgp {
        base routing-protocol-type;
        description
```

```
         "Identity for BGP protocol type.";
    }

    identity static {
        base routing-protocol-type;
        description
         "Identity for static routing protocol type.";
    }

    identity rip {
        base routing-protocol-type;
        description
         "Identity for RIP protocol type.";
    }

    identity rip-ng {
        base routing-protocol-type;
        description
         "Identity for RIPng protocol type.";
    }

    identity vrrp {
        base routing-protocol-type;
        description
         "Identity for VRRP protocol type.
         This is to be used when LAn are directly connected
         to provider Edge routers.";
    }

    identity direct {
        base routing-protocol-type;
        description
         "Identity for direct protocol type.
        .";
    }

    identity protocol-type {
        description
         "Base identity for protocol field type.";
    }

    identity tcp {
        base protocol-type;
        description
         "TCP protocol type.";
    }
    identity udp {
        base protocol-type;
```

```
        description
         "UDP protocol type.";
    }
    identity icmp {
        base protocol-type;
        description
         "icmp protocol type.";
    }
    identity icmp6 {
        base protocol-type;
        description
         "icmp v6 protocol type.";
    }
    identity gre {
        base protocol-type;
        description
         "GRE protocol type.";
    }
    identity ipip {
        base protocol-type;
        description
         "IPinIP protocol type.";
    }
    identity hop-by-hop {
        base protocol-type;
        description
         "Hop by Hop IPv6 header type.";
    }
    identity routing {
        base protocol-type;
        description
         "Routing IPv6 header type.";
    }
    identity esp {
        base protocol-type;
        description
         "ESP header type.";
    }
    identity ah {
        base protocol-type;
        description
         "AH header type.";
    }



    /* Groupings */
```

```
    grouping vpn-service-cloud-access {
        list cloud-access {
            key cloud-identifier;

            leaf cloud-identifier {
                type string;
                description
                 "Identification of cloud service. Local
                 admin meaning.";
            }
            list authorized-sites {
                key site-id;

                leaf site-id {
                    type leafref {
                        path "/l3vpn-svc/sites/site/site-id";
                    }
                    description
                     "Site ID.";
                }
                description
                 "List of authorized sites.";
            }
            list denied-sites {
                key site-id;

                leaf site-id {
                    type leafref {
                        path "/l3vpn-svc/sites/site/site-id";
                    }
                    description
                     "Site ID.";
                }
                description
                 "List of denied sites.";
            }
            leaf nat-enabled {
                type boolean;
                description
                 "Control if NAT is required or not.";
            }
            leaf customer-nat-address {
                type inet:ipv4-address;
                description
                 "NAT address to be used in case of public
                 or shared cloud.
                 This is to be used in case customer is providing
                 the public address.";
```

```
            }
            description
             "Cloud access configuration.";
        }
        description
         "grouping for vpn cloud definition";
    }

    grouping vpn-service-multicast {
        container multicast {
            leaf enabled {
                type boolean;
                default false;
                description
                 "Enable multicast.";
            }
            container customer-tree-flavors {
                list tree-flavor {
                    key type;

                    leaf type {
                        type identityref {
                            base multicast-tree-type;
                        }
                        description
                         "Type of tree to be used.";
                    }
                    description
                     "List of tree flavors.";
                }
                description
                 "Type of trees used by customer.";
            }
            container rp {
                list rp-group-mapping {
                    key "rp-address group";


                    container provider-managed {
                        leaf enabled {
                            type boolean;
                            default false;
                            description
                             "Set to true, if the RP must be a
                             provider
                             managed node.
                             Set to false, if it is a customer
                             managed node.";
```

```
                    }


                    leaf anycast-rp {
                        type boolean;
                        default false;
                        description
                         "Enables anycast-RP.";
                    }
                    description
                     "Parameters for provider managed RP.";
                }


                leaf rp-address {
                    type union {
                        type inet:ipv4-address;
                        type inet:ipv6-address;
                    }
                    description
                    "Defines the address of the RendezvousPoint.
                    Used if RP is customer managed.";
                }
                leaf group {
                    type union {
                        type inet:ipv4-address;
                        type inet:ipv6-address;
                    }
                    description
                    "Defines the address of the multicast group
                    handled by the RP.";
                }
                description
                 "List of RP to group mappings.";
            }
            leaf rp-discovery {
                type identityref {
                    base multicast-rp-discovery-type;
                }
                description
                 "Type of RP discovery used.";
            }

            description
             "RendezvousPoint parameters.";
        }
        description
            "Multicast global parameters for the VPN service.";
```

```
        }
        description
         "grouping for multicast vpn definition";
    }

    grouping vpn-service-mpls {
        leaf mpls {
            type boolean;
            default false;
            description
             "The VPN is using Carrier Supporting Carrier,
             and so MPLS is required.";
        }
        description
         "grouping for mpls CsC definition";
    }


    grouping customer-location-info {
        container location {
                leaf address {
                    type string;
                    description
                    "Address (number and street)
                    of the site.";

                }
                leaf zip-code {
                    type string;
                    description
                    "ZIP code of the site.";
                }
                leaf city {
                    type string;
                    description
                     "City of the site.";
                }
                leaf country-code {
                    type string;
                    description
                     "Country of the site.";
                }
                description
                    "Location of the site.";
        }
        description
         "This grouping defines customer location
          parameters";
```

```
    }

    grouping site-diversity {
        container site-diversity {
                leaf type {
                    type placement-diversity;
                    description
                     "Diversity constraint type.";
                }

                leaf-list site-group {
                    type uint32;
                    description
                     "IDs of group.";
                }
                description
                     "Diversity constraint type.";
            }
        description
         "This grouping defines site diversity
          parameters";
    }
    grouping access-diversity {
        container access-diversity {
                leaf type {
                    type placement-diversity;
                    default "pe-diverse";
                    description
                     "Diversity constraint type.";
                }
                description
                     "Diversity constraint type.";
            }
        description
         "This grouping defines access diversity
          parameters";
    }

    grouping operational-requirements {
          leaf requested-site-start {
               type yang:date-and-time;
               description
                "Optional leaf indicating requested date
                and time
                when the service at a particular site is
                expected
                to start";
           }
```

```
          leaf requested-site-stop {
              type yang:date-and-time;
              description
               "Optional leaf indicating requested date
               and time
               when the service at a particular site is
               expected
               to stop";
          }

          leaf actual-site-start {
              type yang:date-and-time;
              description
               "Optional leaf indicating actual date
               and time
               when the service at a particular site
               actually
               started";
          }
          leaf actual-site-stop {
              type yang:date-and-time;
              description
               "Optional leaf indicating actual date
               and time
               when the service at a particular site
               actually
               stopped";
          }
      description
       "This grouping defines some operational parameters
        parameters";
  }


  grouping flow-definition {
      container match-flow {
          leaf dscp {
              type uint8 {
                  range "0 .. 63";
              }
              description
               "DSCP value.";
          }
          leaf tos {
              type uint8 {
                  range "0 .. 254";
              }
              description
```

```
             "TOS value.";
           }
           leaf dot1p {
               type uint8 {
                   range "0 .. 7";
               }
               description
               "802.1p matching.";
           }
           leaf ipv4-src-prefix {
               type inet:ipv4-prefix;
               description
                "Match on IPv4 src address.";
           }
           leaf ipv6-src-prefix {
               type inet:ipv6-prefix;
               description
                "Match on IPv6 src address.";
           }
           leaf ipv4-dst-prefix {
               type inet:ipv4-prefix;
               description
                "Match on IPv4 dst address.";
           }
           leaf ipv6-dst-prefix {
               type inet:ipv6-prefix;
               description
                "Match on IPv6 dst address.";
           }
           leaf l4-src-port {
               type uint16;
               description
                "Match on layer 4 src port.";
           }
           leaf l4-dst-port {
               type uint16;
               description
                "Match on layer 4 dst port.";
           }
           leaf protocol-field {
               type union {
                   type uint8;
                   type identityref {
                       base protocol-type;
                   }
               }
               description
                "Match on IPv4 protocol or
```

```
                 Ipv6 Next Header
                field.";
            }

            description
             "Describe flow matching
             criterions.";
        }
        description
         "Flow definition based on criteria.";
    }
        grouping site-service-basic {
        leaf svc-input-bandwidth {
               type uint32;
               units bps;
               description
                "From the PE perspective, the service input
                bandwidth of the connection.";
        }
        leaf svc-output-bandwidth {
           type uint32;
           units bps;
           description
            "From the PE perspective, the service output
            bandwidth of the connection.";
        }
        leaf svc-mtu {
            type uint16;
            units bytes;
            description
             "MTU at service level.
             If the service is IP,
             it refers to the IP MTU.";
        }
        description
         "Defines basic service parameters for a site.";
    }
    grouping site-access-protection {
        container traffic-protection {
            leaf enabled {
                type boolean;
                description
                 "Enables
                 traffic protection of access link.";
            }

            description
             "Fast reroute service parameters
```

```
            for the site.";
        }
        description
         "Defines protection service parameters for a site.";
    }
    grouping site-service-mpls {
        container mpls {
            leaf signalling-type {
                type enumeration {
                    enum "ldp" {
                        description
                            "Use LDP as signalling
                            protocol between PE and CE.";
                    }
                    enum "bgp" {
                        description
                            "Use BGP 3107 as signalling
                            protocol between PE and CE.
                            In this case, bgp must be also
                            configured
                            as routing-protocol.
                            ";
                    }
                }
                description
                 "MPLS signalling type.";
            }
            description
             "This container is used when customer provides
             MPLS based services.
             This is used in case of Carrier
             Supporting Carrier.";
        }
        description
         "Defines MPLS service parameters for a site.";
    }
    grouping site-service-qos-profile {
        container qos {
            container qos-classification-policy {
                list rules {
                    key id;
                    ordered-by user;

                    leaf id {
                        type uint16;
                        description
                         "ID of the rule.";
                    }
```

```
                uses flow-definition;

                leaf target-class-id {
                    type string;
                    description
                     "Identification of the
                     class of service.
                     This identifier is internal to
                     the administration.";
                }

                description
                 "List of marking rules.";
            }
            description
             "Need to express marking rules ...";
        }
        container qos-profile {
            choice qos-profile {
                description
                 "Choice for QoS profile.
                 Can be standard profile or custom.";
                case standard {
                    leaf profile {
                        type string;
                        description
                         "QoS profile to be used";
                    }
                }
                case custom {
                    container classes {
                        list class {
                            key class-id;

                            leaf class-id {
                                type string;
                                description
                                 "Identification of the
                                 class of service.
                                 This identifier is internal to
                                 the administration.";
                            }
                            leaf rate-limit {
                                type uint8;
                                units percent;
                                description
                                 "To be used if class must
                                 be rate
```

```
                                    limited. Expressed as
                                    percentage of the svc-bw.";
                                }
                                leaf priority-level {
                                    type uint8;
                                    description
                                     "Defines the level of the
                                     class in
                                     term of priority queueing.
                                      The higher the level is the
                                      higher
                                      is the priority.";
                                }
                                leaf guaranteed-bw-percent {
                                    type uint8;
                                    units percent;
                                    description
                                     "To be used to define the
                                     guaranteed
                                     BW in percent of the svc-bw
                                     available at the priority-level.";
                                }
                                description
                                 "List of class of services.";
                            }
                            description
                                 "Container for
                                 list of class of services.";
                        }

                    }

                }
                description
                "Qos profile configuration.";
            }
            description
             "QoS configuration.";
        }
        description
         "This grouping defines QoS parameters
         for a site";

    }

    grouping site-security-authentication {
        container authentication {
            description
```

```
             "Authentication parameters";
        }
        description
         "This grouping defines authentication
         parameters
         for a site";

    }
    grouping site-security-encryption {
        container encryption {
            leaf enabled {
                type boolean;
                description
                 "If true, access encryption is required.";
            }
            leaf layer {
                type enumeration {
                    enum layer2 {
                        description
                         "Encryption will occur at layer2.";
                    }
                    enum layer3 {
                        description
                         "IPSec is requested.";
                    }
                }
                description
                 "Layer on which encryption is applied.";
            }
            container encryption-profile {
                choice profile {
                    case provider-profile {
                        leaf profile-name {
                            type string;
                            description
                             "Name of the SP profile
                             to be applied.";
                        }
                    }
                    case customer-profile {
                        leaf algorithm {
                            type string;
                            description
                             "Encryption algorithm to
                             be used.";
                        }
                        choice key-type {
                            case psk {
```

```
                                leaf preshared-key {
                                    type string;
                                    description
                                     "Key coming from
                                     customer.";
                                }
                            }
                            case pki {

                            }
                            description
                             "Type of keys to be used.";
                        }
                    }
                    description
                     "Choice of profile.";
                }
                description
                 "Profile of encryption to be applied.";
            }
            description
             "Encryption parameters.";
        }
        description
         "This grouping defines encryption parameters
         for a site";
    }


    grouping site-attachment-bearer {
        container bearer {
            leaf type {
                type string;
                description
                 "Type of bearer Ethernet, DSL, Wireless ...
                 Operator specific.";
            }
            leaf bearer-reference {
                type string;
                description
                 "This is an internal reference for the
                 service provider.";
            }
            description
             "Bearer specific parameters.
             To be augmented.";
        }
        description
```

```
         "Defines physical properties of
         a site attachment.";
    }

    grouping site-routing {
        container routing-protocols {
            list routing-protocol {
                key type;

                leaf type {
                    type identityref {
                        base routing-protocol-type;
                    }
                    description
                     "Type of routing protocol.";
                }


                container ospf {
                    when "../type = 'ospf'" {
                        description
                         "Only applies
                         when protocol is OSPF.";
                    }
                    leaf-list address-family {
                        type identityref {
                            base rt:address-family;
                        }
                        description
                         "Address family to be activated.";
                    }
                    leaf area-address {
                        type yang:dotted-quad;
                        description
                         "Area address.";
                    }
                    leaf metric {
                        type uint16;
                        description
                         "Metric of PE-CE link.";
                    }
                    list sham-link {
                        key target-site;

                        leaf target-site {
                            type svc-id;
                            description
                             "Target site for the sham link
```

```
                    connection.
                    The site is referred through it's ID.";
                }
                leaf metric {
                    type uint16;
                    description
                     "Metric of the sham link.";
                }
                description
                 "Creates a shamlink with another
                 site";
            }
            description
             "OSPF specific configuration.";
        }

        container bgp {
            when "../type = 'bgp'" {
                description
                 "Only applies when
                 protocol is BGP.";
            }
            leaf autonomous-system {
                type uint32;
                description
                 "AS number.";
            }
            leaf-list address-family {
                type identityref {
                    base rt:address-family;
                }
                description
                 "Address family to be activated.";
            }
            description
             "BGP specific configuration.";
        }
        container static {
            when "../type = 'static'" {
                description
                 "Only applies when protocol
                 is static.";
            }

            container cascaded-lan-prefixes {
                list ipv4-lan-prefixes {
                    key "lan next-hop";
```

```
                    leaf lan {
                        type inet:ipv4-prefix;
                        description
                         "Lan prefixes.";
                    }
                    leaf lan-tag {
                        type string;
                        description
                         "Internal tag to be used in vpn
                         policies.";
                    }
                    leaf next-hop {
                        type inet:ipv4-address;
                        description
                         "Nexthop address to use at customer
                         side.";
                    }
                    description "
                        List of LAN prefixes for
                        the site.
                        ";
                }
                list ipv6-lan-prefixes {
                    key "lan next-hop";

                    leaf lan {
                        type inet:ipv6-prefix;
                        description
                         "Lan prefixes.";
                    }
                    leaf lan-tag {
                        type string;
                        description
                         "Internal tag to be used
                         in vpn policies.";
                    }
                    leaf next-hop {
                        type inet:ipv6-address;
                        description
                         "Nexthop address to use at
                         customer side.";
                    }
                    description "
                        List of LAN prefixes for the site.
                        ";
                }
                description
                    "LAN prefixes from the customer.";
```

```
                    }
                    description
                     "Static routing
                     specific configuration.";
                }
                container rip {
                    when "../type = 'rip'" {
                        description
                         "Only applies when
                         protocol is RIP.";
                    }
                    leaf-list address-family {
                        type identityref {
                            base rt:address-family;
                        }
                        description
                         "Address family to be
                         activated.";
                    }

                    description
                     "RIP routing specific
                     configuration.";
                }


                container vrrp {
                    when "../type = 'vrrp'" {
                        description
                         "Only applies when
                         protocol is VRRP.";
                    }
                    leaf-list address-family {
                        type identityref {
                            base rt:address-family;
                        }
                        description
                         "Address family to be activated.";
                    }
                    description
                     "VRRP routing specific configuration.";
                }


                description
                 "List of routing protocols used
                 on the site.
                 Need to be augmented.";
```

```
            }
            description
             "Defines routing protocols.";
        }
        description
         "Grouping for routing protocols.";
    }

    grouping site-attachment-ip-connection {
        container ip-connection {
            container ipv4 {
                leaf address-allocation-type {
                    type identityref {
                        base address-allocation-type;
                    }
                    description
                     "Defines how addresses are allocated.
                     Need to be detailed further.";
                }
                choice subnet {
                    case subnet-only {
                        leaf subnet-prefix {
                            type inet:ipv4-prefix;
                            description
                             "Interco subnet.";
                        }
                    }
                    case addresses {
                        leaf provider-address {
                            type inet:ipv4-address;
                            description
                             "Provider side address.";
                        }
                        leaf customer-address {
                            type inet:ipv4-address;
                            description
                             "Customer side address.";
                        }
                        leaf mask {
                            type uint8 {
                                range "0..32";
                            }
                            description
                             "Subnet mask expressed
                             in bits";
                        }
                    }
                    description
```

```
                    "Choice for addressing
                    customer to network link.";
                }

                description
                 "IPv4 specific parameters";

            }
            container ipv6 {
                leaf address-allocation-type {
                    type string;
                    description
                     "Defines how addresses are allocated.
                     Need to be detailled further.";
                }
                choice subnet {
                    case subnet-only {
                        leaf subnet-prefix {
                            type inet:ipv6-prefix;
                            description
                             "Interco subnet.";
                        }
                    }
                    case addresses {
                        leaf provider-address {
                            type inet:ipv6-address;
                            description
                             "Provider side address.";
                        }
                        leaf customer-address {
                            type inet:ipv6-address;
                            description
                             "Customer side address.";
                        }
                        leaf mask {
                            type uint8 {
                                range "0..128";
                            }
                            description
                             "Subnet mask expressed
                             in bits";
                        }
                    }
                    description
                     "Choice for addressing
                     customer to network link.";
                }
                description
```

```
                "IPv6 specific parameters";

            }
            container oam {
                container bfd {
                    leaf bfd-enabled {
                        type boolean;
                        description
                         "BFD activation";
                    }

                    choice holdtime {
                        case profile {
                            leaf profile-name {
                                type string;
                                description
                                 "Service provider well
                                 known profile.";
                            }
                            description
                                 "Service provider well
                                 known profile.";
                        }
                        case fixed {
                            leaf fixed-value {
                                type uint32;
                                units msec;
                                description
                                 "Expected holdtime
                                 expressed
                                 in msec.";
                            }
                        }
                        description
                         "Choice for holdtime flavor.";
                    }
                    description
                     "Container for BFD.";
                }
                description
                 "Define the OAM used on the connection.";
            }
            description
             "Defines connection parameters.";
        }
        description
         "This grouping defines IP connection parameters.";
    }
```

```
    grouping site-service-multicast {
        container multicast {
            leaf multicast-site-type {
                type enumeration {
                    enum receiver-only {
                        description
                         "The site has only receivers.";
                    }
                    enum source-only {
                        description
                         "The site has only sources.";
                    }
                    enum source-receiver {
                        description
                         "The site has both
                         sources & receivers.";
                    }
                }
                default "source-receiver";
                description
                 "Type of multicast site.";
            }
            container multicast-transport-protocol {
                leaf ipv4 {
                    type boolean;
                    default true;
                    description
                        "Enables ipv4 multicast transport";
                }
                leaf ipv6 {
                    type boolean;
                    default false;
                    description
                        "Enables ipv6 multicast transport";
                }
                description
                 "Defines protocol to transport multicast.";
            }
            leaf protocol-type {
                type enumeration {
                    enum host {
                        description
                         "
                         Hosts are directly connected
                         to the provider network.
                         Host protocols like IGMP, MLD
                         are required.
                         ";
```

```
                    }
                    enum router {
                        description
                         "
                         Hosts are behind a customer router.
                         PIM will be implemented.
                         ";
                    }
                    enum both {
                        description
                         "Some Hosts are behind a customer
                         router and some others are directly
                         connected to the provider network.
                         Both host and routing protocols must be
                         used. Typically IGMP and PIM will be
                         implemented.
                         ";
                    }
                }
                default "both";
                description
                 "Multicast protocol type to be used
                 with the customer site.";
            }

            description
             "Multicast parameters for the site.";
        }
        description
          "Multicast parameters for the site.";
    }

    grouping site-management {
        container management {
            leaf type {
                type identityref {
                    base management;
                }
                description
                "Management type of the connection.";
            }
            leaf management-transport {
                type identityref {
                    base rt:address-family;
                }
                description
                 "Transport protocol used for management.";
            }
```

```
            leaf address {
                type union {
                    type inet:ipv4-address;
                    type inet:ipv6-address;
                }
                description
                 "Management address";
            }

            description
             "Management configuration";
        }
        description
          "Management parameters for the site.";
    }

    grouping site-vpn-flavor {
        leaf site-vpn-flavor {
            type identityref {
                base site-vpn-flavor;
            }
            default site-vpn-flavor-single;
            description
             "Defines if the site
            is a single VPN site, or multiVPN or ...";
        }
        description
         "Grouping for site-vpn-flavor.";
    }

    grouping site-vpn-policy {
        container vpn-policy-list {
            list vpn-policy {
                key vpn-policy-id;

                leaf vpn-policy-id {
                    type svc-id;
                    description
                     "Unique identifier for
                     the VPN policy.";
                }

                list entries {
                    key id;

                    leaf id {
                            type svc-id;
                            description
```

```
                "Unique identifier for
                 the policy entry.";
            }
            container filter {
                choice lan {
                    case lan-prefix {
                        container lan-prefixes {
                            list ipv4-lan-prefixes {
                                key lan;

                                leaf lan {
                                    type inet:ipv4-prefix;
                                    description
                                     "Lan prefixes.";
                                }
                                description "
                                    List of LAN prefixes
                                    for the site.
                                    ";
                            }
                            list ipv6-lan-prefixes {
                                key lan;

                                leaf lan {
                                    type inet:ipv6-prefix;
                                    description
                                     "Lan prefixes.";
                                }
                                description "
                                    List of LAN prefixes
                                    for the site.
                                    ";
                            }
                            description
                             "LAN prefixes from the customer.";
                        }
                    }
                    case lan-tag {
                        leaf-list lan-tag {
                            type string;
                            description
                             "List of lan-tags to be matched.";
                        }
                    }
                    description
                     "Choice for LAN matching type";
                }
                description
```

```
                        "If used, it permit to split site LANs
                        among multiple VPNs.
                        If no filter used, all the LANs will be
                        part of the same VPNs with the same
                        role.";
                    }
                    container vpn {
                        leaf vpn-id {
                            type leafref {
                                path "/l3vpn-svc/vpn-services/vpn-svc/
vpn-id";
                            }
                            mandatory true;
                            description
                             "Reference to an IPVPN.";
                        }
                        leaf site-role {
                            type identityref {
                                base site-role;
                            }
                            mandatory true;
                            description
                             "Role of the site in the IPVPN.";
                        }
                        description
                         "List of VPNs the LAN is associated to.";
                    }
                    description
                     "List of entries for export policy.";
                }
                description
                 "List of VPN policies.";
            }
            description
                 "VPN policy.";
        }
        description
          "VPN policy parameters for the site.";
    }



    grouping site-maximum-routes {
        container maximum-routes {
            list address-family {
                key af;

                leaf af {
                    type identityref {
```

```
                    base rt:address-family;
                }
                description
                 "Address-family.";
            }
            leaf maximum-routes {
                type uint32;
                description
                 "Maximum prefixes the VRF can
                 accept for this
                 address-family.";
            }
            description
             "List of address families.";
        }

        description
         "Define maximum-routes for the VRF.";
    }
    description
    "Define maximum-routes for the site.";
}

grouping site-security {
    container security {
        uses site-security-authentication;
        uses site-security-encryption;

        description
         "Site specific security parameters.";
    }
    description
     "Grouping for security parameters.";
}

grouping site-service {
    container service {
        uses site-service-basic;
        uses site-service-qos-profile;
        uses site-service-mpls;
        uses site-service-multicast;

        description
         "Service parameters on the attachement.";
    }
    description
     "Grouping for service parameters.";
}
```

```
    grouping transport-constraint {
        list constraint-list {
            key constraint-type;

            leaf constraint-type {
                type identityref {
                    base transport-constraint;
                }
                description
                 "Constraint type to be applied.";
            }
            leaf constraint-opaque-value {
                type string;
                description
                "Opaque value that can be used to
                specify constraint parameters.";
            }
            description
             "List of constraints";
        }
        description
         "Grouping for transport constraint.";
    }

    grouping transport-constraints {
        container transport-constraints {
            container unicast-transport-constraints {
                list constraints {
                    key constraint-id;

                    leaf constraint-id {
                        type svc-id;
                        description
                         "Defines an ID for the constraint
                         rule.";
                    }

                    leaf site1 {
                        type svc-id;
                        description
                         "The ID refers to one site end.";
                    }
                    leaf site2 {
                        type svc-id;
                        description
                         "The ID refers to the other
                         site end.";
                    }
```

```
                    uses transport-constraint;
                    description
                     "List of constraints.
                     Constraints are bidirectional.";
                }
                description
                 "Unicast transport constraints.";
            }
            container multicast-transport-constraints {
                list constraints {
                    key constraint-id;

                    leaf constraint-id {
                        type svc-id;
                        description
                         "Defines an ID for the constraint
                         rule.";
                    }

                    leaf src-site {
                        type svc-id;
                        description
                         "The ID refers to source site.";
                    }
                    leaf dst-site {
                        type svc-id;
                        description
                         "The ID refers to the receiver
                         site.";
                    }
                    uses transport-constraint;
                    description
                     "List of constraints.
                     Constraints are unidirectional.";
                }
                description
                 "Multicast transport constraints.";
            }
            description
                 "transport constraints.";
        }
        description
         "Grouping for transport constraints
         description.";
    }

    grouping vpn-extranet {
        container extranet-vpns {
```

```
            list extranet-vpn {
                key vpn-id;

                leaf vpn-id {
                    type svc-id;
                    description
                        "Identifies the target VPN";
                }
                leaf local-sites-role {
                    type identityref {
                        base site-role;
                    }
                    description
                     "This describes the role of the
                     local sites in the target VPN topology.";
                }
                description
                 "List of extranet VPNs the local
                 VPN is attached to.";
            }
            description
             "Container for extranet vpn cfg.";
        }
        description
            "grouping for extranet VPN configuration.
            Extranet provides a way to interconnect all sites
            from two VPNs in a easy way.";

    }

    grouping site-attachment-availability {
        container availability {
            uses site-access-protection;
            leaf access-priority {
                type uint32;
                default 1;
                description
                 "Defines the priority for the access.
                 The highest the priority value is,
                 the highest the
                 preference of the access is.";
            }
            description
             "Availability parameters
             (used for multihoming)";
        }
        description
         "Defines site availability parameters.";
```

```
    }

    grouping access-vpn-policy {
        container vpn-attachment {

            choice attachment-flavor {
                case vpn-policy-id {
                    leaf vpn-policy-id {
                        type leafref {
                            path "/l3vpn-svc/sites/site/"+
                            "vpn-policy-list/vpn-policy/"+
                            "vpn-policy-id";
                        }
                        description
                         "Reference to a VPN policy.";
                    }
                }
                case vpn-id {
                    leaf vpn-id {
                        type leafref {
                            path "/l3vpn-svc/vpn-services"+
                            "/vpn-svc/vpn-id";
                        }
                        description
                            "Reference to a VPN.";
                    }
                    leaf site-role {
                        type identityref {
                                base site-role;
                            }
                        mandatory true;
                        description
                         "Role of the site in the IPVPN.";
                    }
                }
                mandatory true;
                description
                 "Choice for VPN attachment flavor.";
            }
            description
             "Defines VPN attachment of a site.";
        }
        description
         "Defines the VPN attachment rules
         for a site logical access.";
    }

    grouping vpn-svc-cfg {
```

```
        leaf vpn-id {
                type svc-id;
                description
                "VPN identifier. Local administration meaning.";
            }
        leaf customer-name {
            type string;
            description
             "Name of the customer.";
        }
        leaf topology {
            type identityref {
                base vpn-topology;
            }
            default "any-to-any";
            description
             "VPN topology.";
        }

        uses vpn-service-cloud-access;
        uses vpn-service-multicast;
        uses vpn-service-mpls;
        uses transport-constraints;
        uses vpn-extranet;

        description
         "grouping for vpn-svc configuration.";
    }

    grouping site-top-level-cfg {
        uses operational-requirements;
        uses customer-location-info;
        uses site-diversity;
        uses site-management;
        uses site-vpn-policy;
        uses site-vpn-flavor;
        uses site-maximum-routes;
        uses site-security;
        uses site-service;
        uses site-routing;

        description
         "Grouping for site top level cfg.";
    }
    grouping site-network-access-top-level-cfg {
        uses access-diversity;
        uses site-attachment-bearer;
        uses site-attachment-ip-connection;
```

```
        uses site-security;
        uses site-service;
        uses site-routing;
        uses site-attachment-availability;
        uses access-vpn-policy;

        description
         "Grouping for site network access
         top level cfg.";
    }

    /* Main blocks */

    container l3vpn-svc {
        container vpn-services {
            list vpn-svc {
                key vpn-id;

                uses vpn-svc-cfg;

                description "
                    List of VPN services.
                ";
            }
            description
             "top level container
             for the VPN services.";
        }

        container sites {
            list site {
                key site-id;

                leaf site-id {
                    type svc-id;
                    description
                        "Identifier of the site.";
                }

                leaf apply-template {
                    type leafref {
                        path "/l3vpn-svc/site-templates"+
                        "/site-template/site-template-id";
                    }
                    description
                     "Reference to template to be applied.
                     The template is called through it's ID.";
                }
```

```
            uses site-top-level-cfg;

            container site-network-accesses {
                list site-network-access {
                    key site-network-access-id;

                    leaf site-network-access-id {
                        type svc-id;
                        description
                         "Identifier for the access";
                    }
                    leaf apply-template {
                        type leafref {
                            path "/l3vpn-svc/site-templates/"+
                            "site-template/site-template-id";
                        }
                        description
                         "Reference to template to be applied.
                         The template is called through it's ID.";
                    }
                    uses site-network-access-top-level-cfg;

                    description
                     "List of accesses for a site.";
                }
                description
                 "List of accesses for a site.";
            }

            description "List of sites.";
        }
        description
         "Container for sites";
    }

    container site-templates {
        list site-template {
            key site-template-id;

            leaf site-template-id {
                type template-id;
                description
                    "Identifier of the site.";
            }

            uses site-top-level-cfg;

            container site-network-access {
```

```
                    uses site-network-access-top-level-cfg;
                    description
                     "Container for the site-network-access.";
                }

                description "List of sites.";
            }
            description "Container for site templates";
        }


        description
         "Main container for L3VPN service configuration.";
    }

}
<CODE ENDS>
```

9. Security Considerations

TBD.

10. Acknowledgements

Thanks to Qin Wu, Maxim Klyus, Luis Miguel Contreras, Gregory Mirsky,
Zitao Wang, Jing Zhao, Kireeti Kompella, Eric Rosen, Aijun Wang,
Kenichi Ogaki and Andrew Leu for the contributions to the document.

11. IANA Considerations

TBD.

12. References

12.1. Normative References

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

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <http://www.rfc-editor.org/info/rfc4364>.

[RFC4577] Rosen, E., Psenak, P., and P. Pillay-Esnault, "OSPF as the Provider/Customer Edge Protocol for BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4577, DOI 10.17487/RFC4577, June 2006, <http://www.rfc-editor.org/info/rfc4577>.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <http://www.rfc-editor.org/info/rfc4862>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <http://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, <http://www.rfc-editor.org/info/rfc6241>.

[RFC6513] Rosen, E., Ed. and R. Aggarwal, Ed., "Multicast in MPLS/BGP IP VPNs", RFC 6513, DOI 10.17487/RFC6513, February 2012, <http://www.rfc-editor.org/info/rfc6513>.

12.2. Informative References

[RFC4110] Callon, R. and M. Suzuki, "A Framework for Layer 3 Provider-Provisioned Virtual Private Networks (PPVPNs)", RFC 4110, DOI 10.17487/RFC4110, July 2005, <http://www.rfc-editor.org/info/rfc4110>.

Appendix A. Example: NETCONF <get> Reply

This section gives an example of a reply to the NETCONF <get> request for a device that implements the data model defined in this document.

The example is written in XML.

Authors' Addresses

Stephane Litkowski
Orange Business Service

Email: stephane.litkowski@orange.com

Rob Shakir
BT

Email: rob.shakir@bt.com

Luis Tomotaki
Verizon

Email: luis.tomotaki@verizon.com

Kenichi Ogaki
KDDI

Email: ke-oogaki@kddi.com

Kevin D'Souza
ATT

Email: kd6913@att.com