L3SM Working Group   S. Litkowski
Internet-Draft   Orange Business Services
Intended status: Standards Track   R. Shakir
Expires: December 11, 2016   Jive Communications
                                                             L. Tomotaki
                                                                 Verizon
                                                                K. Ogaki
                                                                    KDDI
                                                              K. D'Souza
                                                                     ATT
                                                           June 09, 2016

YANG Data Model for L3VPN service delivery
draft-ietf-l3sm-l3vpn-service-model-08

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

Copyright Notice

Table of Contents

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: Equipment that is dedicated to a particular customer and is directly connected (at layer 3) to one or more PE devices via attachment circuits. A CE is usually located at the customer premises, and is usually dedicated to a single VPN, although it may support multiple VPNs if each one has separate attachment circuits.

Provider Edge (PE) Device: Equipment managed by the SP that can support multiple VPNs for different customers, and is directly connected (at layer 3) to one or more CE devices via attachment circuits. A PE is usually located at an SP point of presence (PoP) and is managed by the SP.

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

```
            L3VPN-SVC |
              MODEL   |
                      |
                   +------------------+         +-----+
                   |   Orchestration  | < --- > | OSS |
                   +------------------+         +-----+
                      |           |
              +----------------+  |
              | Config manager |  |
              +----------------+  |
                      |           |
                      | Netconf/CLI ...
                      |           |
        +------------------------------------------------+
                          Network

                        +++++++
                        + AAA +
                        +++++++

   +++++++  Bearer ++++++++             ++++++++       +++++++
   + CEA + ------- + PE A +             + PE B + ----- + CEB +
   +++++++  Cnct   ++++++++             ++++++++       +++++++

   Site A                                       Site B
```

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-services, sites, site-templates.

The vpn-svc under vpn-services 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 properties of the attachment that are below layer 3 while the connection refers to layer 3 protocol oriented properties. The bearer may be allocated dynamically by the service provider and the customer may provide some constraints or parameters to drive the placement.

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-accesses
      |     |  +--rw cloud-access* [cloud-identifier] {cloud-access}?
      |     |     +--rw cloud-identifier        string
      |     |     +--rw authorized-sites
      |     |     |  +--rw authorized-site* [site-id]
      |     |     |     +--rw site-id    leafref
      |     |     +--rw denied-sites
      |     |     |  +--rw denied-site* [site-id]
      |     |     |     +--rw site-id    leafref
      |     |     +--rw nat-enabled?            boolean
      |     |     +--rw customer-nat-address?   inet:ipv4-address
      |     +--rw multicast {multicast}?
      |     |  +--rw enabled?                 boolean
      |     |  +--rw customer-tree-flavors
      |     |  |  +--rw tree-flavor* [type]
      |     |  |     +--rw type    identityref
```

```
      |     |  +--rw rp
      |     |     +--rw rp-group-mappings
      |     |     |  +--rw rp-group-mapping* [id]
      |     |     |     +--rw id                  uint16
      |     |     |     +--rw provider-managed
      |     |     |     |  +--rw enabled?                    boolean
      |     |     |     |  +--rw rp-redundancy?              boolean
      |     |     |     |  +--rw optimal-traffic-delivery?   boolean
      |     |     |     +--rw rp-address?         inet:ip-address
      |     |     |     +--rw groups
      |     |     |        +--rw group* [id]
      |     |     |           +--rw id               uint16
      |     |     |           +--rw (group-format)?
      |     |     |              +--:(startend)
      |     |     |              |  +--rw group-start?     inet:ip-address
      |     |     |              |  +--rw group-end?       inet:ip-address
      |     |     |              +--:(singleaddress)
      |     |     |                 +--rw group-address?   inet:ip-address
      |     |     +--rw rp-discovery
      |     |        +--rw rp-discovery-type?   identityref
      |     |        +--rw bsr-candidates
      |     |           +--rw bsr-candidate* [address]
      |     |              +--rw address    inet:ip-address
      |     +--rw carrierscarrier?         boolean {carrierscarrier}?
      |     +--rw transport-constraints {traffic-engineering}?
      |     |  +--rw unicast-transport-constraints
      |     |  |  +--rw constraint* [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 {traffic-engineering-
multicast}?
      |     |     +--rw constraint* [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 {extranet-vpn}?
      |        +--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 location
      |     |  +--rw address?        string
      |     |  +--rw zip-code?       string
      |     |  +--rw state?          string
      |     |  +--rw city?           string
      |     |  +--rw country-code?   string
      |     +--rw site-diversity {site-diversity}?
      |     |  +--rw groups
      |     |     +--rw group* [group-id]
      |     |        +--rw group-id    string
      |     +--rw management
      |     |  +--rw type?                   identityref
      |     |  +--rw management-transport?   identityref
      |     |  +--rw address?                inet:ip-address
      |     +--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)?
      |     |        |     +--:(lan-prefix)
      |     |        |     |  +--rw lan-prefixes
      |     |        |     |     +--rw ipv4-lan-prefixes* [lan] {ipv4}?
      |     |        |     |     |  +--rw lan    inet:ipv4-prefix
      |     |        |     |     +--rw ipv6-lan-prefixes* [lan] {ipv6}?
      |     |        |     |        +--rw lan    inet:ipv6-prefix
      |     |        |     +--:(lan-tag)
      |     |        |        +--rw lan-tag*        string
      |     |        +--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 {encryption}?
      |     |     +--rw enabled?              boolean
      |     |     +--rw layer?                enumeration
      |     |     +--rw encryption-profile
      |     |        +--rw (profile)?
      |     |           +--:(provider-profile)
```

```
      |     |           |  +--rw profile-name?    string
      |     |           +--:(customer-profile)
      |     |              +--rw algorithm?       string
      |     |              +--rw (key-type)?
      |     |                 +--:(psk)
      |     |                 |  +--rw preshared-key?   string
      |     |                 +--:(pki)
      |     +--rw service
      |     |  +--rw svc-input-bandwidth?    uint32
      |     |  +--rw svc-output-bandwidth?   uint32
      |     |  +--rw svc-mtu?                uint16
      |     |  +--rw qos {qos}?
      |     |  |  +--rw qos-classification-policy
      |     |  |  |  +--rw rule* [id]
      |     |  |  |     +--rw id                   uint16
      |     |  |  |     +--rw (match-type)?
      |     |  |  |     |  +--:(match-flow)
      |     |  |  |     |  |  +--rw match-flow
      |     |  |  |     |  |     +--rw dscp?              uint8
      |     |  |  |     |  |     +--rw tos?               uint8
      |     |  |  |     |  |     +--rw dot1p?             uint8
      |     |  |  |     |  |     +--rw ipv4-src-prefix?   inet:ipv4-prefix
      |     |  |  |     |  |     +--rw ipv6-src-prefix?   inet:ipv6-prefix
      |     |  |  |     |  |     +--rw ipv4-dst-prefix?   inet:ipv4-prefix
      |     |  |  |     |  |     +--rw ipv6-dst-prefix?   inet:ipv6-prefix
      |     |  |  |     |  |     +--rw l4-src-port?       uint16
      |     |  |  |     |  |     +--rw l4-dst-port?       uint16
      |     |  |  |     |  |     +--rw protocol-field?    union
      |     |  |  |     |  +--:(match-application)
      |     |  |  |     |     +--rw match-application?   identityref
      |     |  |  |     +--rw target-class-id?    string
      |     |  |  +--rw qos-profile
      |     |  |     +--rw (qos-profile)?
      |     |  |        +--:(standard)
      |     |  |        |  +--rw profile?   string
      |     |  |        +--:(custom)
      |     |  |           +--rw classes {qos-custom}?
      |     |  |              +--rw class* [class-id]
      |     |  |                 +--rw class-id                 string
      |     |  |                 +--rw rate-limit?              uint8
      |     |  |                 +--rw priority-level?          uint8
      |     |  |                 +--rw guaranteed-bw-percent?   uint8
      |     |  +--rw carrierscarrier {carrierscarrier}?
      |     |  |  +--rw signalling-type?   enumeration
      |     |  +--rw multicast {multicast}?
      |     |     +--rw multicast-site-type?            enumeration
      |     |     +--rw multicast-transport-protocol
      |     |     |  +--rw ipv4?   boolean {ipv4}?
```

```
      |     |     |  +--rw ipv6?   boolean {ipv6}?
      |     |     +--rw protocol-type?                  enumeration
      |     +--rw traffic-protection {fast-reroute}?
      |     |  +--rw enabled?   boolean
      |     +--rw routing-protocols
      |     |  +--rw routing-protocol* [type]
      |     |     +--rw type      identityref
      |     |     +--rw ospf {rtg-ospf}?
      |     |     |  +--rw address-family*   identityref
      |     |     |  +--rw area-address?     yang:dotted-quad
      |     |     |  +--rw metric?           uint16
      |     |     |  +--rw sham-links {rtg-ospf-sham-link}?
      |     |     |     +--rw sham-link* [target-site]
      |     |     |        +--rw target-site    svc-id
      |     |     |        +--rw metric?        uint16
      |     |     +--rw bgp {rtg-bgp}?
      |     |     |  +--rw autonomous-system?   uint32
      |     |     |  +--rw address-family*      identityref
      |     |     +--rw static
      |     |     |  +--rw cascaded-lan-prefixes
      |     |     |     +--rw ipv4-lan-prefixes* [lan next-hop] {ipv4}?
      |     |     |     |  +--rw lan         inet:ipv4-prefix
      |     |     |     |  +--rw lan-tag?    string
      |     |     |     |  +--rw next-hop    inet:ipv4-address
      |     |     |     +--rw ipv6-lan-prefixes* [lan next-hop] {ipv6}?
      |     |     |        +--rw lan         inet:ipv6-prefix
      |     |     |        +--rw lan-tag?    string
      |     |     |        +--rw next-hop    inet:ipv6-address
      |     |     +--rw rip {rtg-rip}?
      |     |     |  +--rw address-family*   identityref
      |     |     +--rw vrrp {rtg-vrrp}?
      |     |        +--rw address-family*   identityref
      |     +--ro actual-site-start?       yang:date-and-time
      |     +--ro actual-site-stop?        yang:date-and-time
      |     +--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 {site-diversity}?
      |           |  +--rw groups
      |           |  |  +--rw group* [group-id]
      |           |  |     +--rw group-id    string
      |           |  +--rw constraints
      |           |     +--rw constraint* [constraint-type]
      |           |        +--rw constraint-type    identityref
      |           |        +--rw target
      |           |           +--rw (target-flavor)?
      |           |              +--:(id)
```

```
      |           |              |  +--rw group* [group-id]
      |           |              |     +--rw group-id    string
      |           |              +--:(all-accesses)
      |           |              |  +--rw all-other-accesses?   empty
      |           |              +--:(all-groups)
      |           |                 +--rw all-other-groups?     empty
      |           +--rw bearer
      |           |  +--rw requested-type {requested-type}?
      |           |  |  +--rw requested-type?   string
      |           |  |  +--rw strict?           boolean
      |           |  +--rw always-on?          boolean {always-on}?
      |           |  +--rw bearer-reference?   string {bearer-reference}?
      |           +--rw ip-connection
      |           |  +--rw ipv4 {ipv4}?
      |           |  |  +--rw address-allocation-type?   identityref
      |           |  |  +--rw addresses
      |           |  |     +--rw provider-address?   inet:ipv4-address
      |           |  |     +--rw customer-address?   inet:ipv4-address
      |           |  |     +--rw mask?               uint8
      |           |  +--rw ipv6 {ipv6}?
      |           |  |  +--rw address-allocation-type?   identityref
      |           |  |  +--rw addresses
      |           |  |     +--rw provider-address?   inet:ipv6-address
      |           |  |     +--rw customer-address?   inet:ipv6-address
      |           |  |     +--rw mask?               uint8
      |           |  +--rw oam
      |           |     +--rw bfd {bfd}?
      |           |        +--rw bfd-enabled?    boolean
      |           |        +--rw (holdtime)?
      |           |           +--:(profile)
      |           |           |  +--rw profile-name?   string
      |           |           +--:(fixed)
      |           |              +--rw fixed-value?    uint32
      |           +--rw security
      |           |  +--rw authentication
      |           |  +--rw encryption {encryption}?
      |           |     +--rw enabled?              boolean
      |           |     +--rw layer?                enumeration
      |           |     +--rw encryption-profile
      |           |        +--rw (profile)?
      |           |           +--:(provider-profile)
      |           |           |  +--rw profile-name?    string
      |           |           +--:(customer-profile)
      |           |              +--rw algorithm?       string
      |           |              +--rw (key-type)?
      |           |                 +--:(psk)
      |           |                 |  +--rw preshared-key?   string
      |           |                 +--:(pki)
```

```
      |           +--rw service
      |           |  +--rw svc-input-bandwidth?    uint32
      |           |  +--rw svc-output-bandwidth?   uint32
      |           |  +--rw svc-mtu?                uint16
      |           |  +--rw qos {qos}?
      |           |  |  +--rw qos-classification-policy
      |           |  |  |  +--rw rule* [id]
      |           |  |  |     +--rw id                   uint16
      |           |  |  |     +--rw (match-type)?
      |           |  |  |     |  +--:(match-flow)
      |           |  |  |     |  |  +--rw match-flow
      |           |  |  |     |  |     +--rw dscp?              uint8
      |           |  |  |     |  |     +--rw tos?               uint8
      |           |  |  |     |  |     +--rw dot1p?             uint8
      |           |  |  |     |  |     +--rw ipv4-src-prefix?   inet:ipv4-
prefix
      |           |  |  |     |  |     +--rw ipv6-src-prefix?   inet:ipv6-
prefix
      |           |  |  |     |  |     +--rw ipv4-dst-prefix?   inet:ipv4-
prefix
      |           |  |  |     |  |     +--rw ipv6-dst-prefix?   inet:ipv6-
prefix
      |           |  |  |     |  |     +--rw l4-src-port?       uint16
      |           |  |  |     |  |     +--rw l4-dst-port?       uint16
      |           |  |  |     |  |     +--rw protocol-field?    union
      |           |  |  |     |  +--:(match-application)
      |           |  |  |     |     +--rw match-application?   identityref
      |           |  |  |     +--rw target-class-id?     string
      |           |  |  +--rw qos-profile
      |           |  |     +--rw (qos-profile)?
      |           |  |        +--:(standard)
      |           |  |        |  +--rw profile?   string
      |           |  |        +--:(custom)
      |           |  |           +--rw classes {qos-custom}?
      |           |  |              +--rw class* [class-id]
      |           |  |                 +--rw class-id                 string
      |           |  |                 +--rw rate-limit?              uint8
      |           |  |                 +--rw priority-level?          uint8
      |           |  |                 +--rw guaranteed-bw-percent?   uint8
      |           |  +--rw carrierscarrier {carrierscarrier}?
      |           |  |  +--rw signalling-type?   enumeration
      |           |  +--rw multicast {multicast}?
      |           |     +--rw multicast-site-type?            enumeration
      |           |     +--rw multicast-transport-protocol
      |           |     |  +--rw ipv4?   boolean {ipv4}?
      |           |     |  +--rw ipv6?   boolean {ipv6}?
      |           |     +--rw protocol-type?                  enumeration
      |           +--rw routing-protocols
```

```
   |           |  +--rw routing-protocol* [type]
   |           |     +--rw type      identityref
   |           |     +--rw ospf {rtg-ospf}?
   |           |     |  +--rw address-family*   identityref
```

```
      |           |     |  +--rw area-address?     yang:dotted-quad
      |           |     |  +--rw metric?           uint16
      |           |     |  +--rw sham-links {rtg-ospf-sham-link}?
      |           |     |     +--rw sham-link* [target-site]
      |           |     |        +--rw target-site    svc-id
      |           |     |        +--rw metric?        uint16
      |           |     +--rw bgp {rtg-bgp}?
      |           |     |  +--rw autonomous-system?   uint32
      |           |     |  +--rw address-family*      identityref
      |           |     +--rw static
      |           |     |  +--rw cascaded-lan-prefixes
      |           |     |     +--rw ipv4-lan-prefixes* [lan next-hop] {ipv4}?
      |           |     |     |  +--rw lan         inet:ipv4-prefix
      |           |     |     |  +--rw lan-tag?    string
      |           |     |     |  +--rw next-hop    inet:ipv4-address
      |           |     |     +--rw ipv6-lan-prefixes* [lan next-hop] {ipv6}?
      |           |     |        +--rw lan         inet:ipv6-prefix
      |           |     |        +--rw lan-tag?    string
      |           |     |        +--rw next-hop    inet:ipv6-address
      |           |     +--rw rip {rtg-rip}?
      |           |     |  +--rw address-family*   identityref
      |           |     +--rw vrrp {rtg-vrrp}?
      |           |        +--rw address-family*   identityref
      |           +--rw availability
      |           |  +--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
      +--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 location
            |  +--rw address?        string
            |  +--rw zip-code?       string
            |  +--rw state?          string
            |  +--rw city?           string
            |  +--rw country-code?   string
            +--rw site-diversity {site-diversity}?
            |  +--rw groups
            |     +--rw group* [group-id]
            |        +--rw group-id    string
            +--rw management
```

```
            |  +--rw type?                   identityref
            |  +--rw management-transport?   identityref
            |  +--rw address?                inet:ip-address
            +--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)?
            |        |     +--:(lan-prefix)
            |        |     |  +--rw lan-prefixes
            |        |     |     +--rw ipv4-lan-prefixes* [lan] {ipv4}?
            |        |     |     |  +--rw lan    inet:ipv4-prefix
            |        |     |     +--rw ipv6-lan-prefixes* [lan] {ipv6}?
            |        |     |        +--rw lan    inet:ipv6-prefix
            |        |     +--:(lan-tag)
            |        |        +--rw lan-tag*        string
            |        +--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 {encryption}?
            |     +--rw enabled?              boolean
            |     +--rw layer?                enumeration
            |     +--rw encryption-profile
            |        +--rw (profile)?
            |           +--:(provider-profile)
            |           |  +--rw profile-name?    string
            |           +--:(customer-profile)
            |              +--rw algorithm?       string
            |              +--rw (key-type)?
            |                 +--:(psk)
            |                 |  +--rw preshared-key?   string
            |                 +--:(pki)
            +--rw service
            |  +--rw svc-input-bandwidth?    uint32
            |  +--rw svc-output-bandwidth?   uint32
            |  +--rw svc-mtu?                uint16
            |  +--rw qos {qos}?
            |  |  +--rw qos-classification-policy
            |  |  |  +--rw rule* [id]
```

```
            |  |  |     +--rw id                   uint16
            |  |  |     +--rw (match-type)?
            |  |  |     |  +--:(match-flow)
            |  |  |     |  |  +--rw match-flow
            |  |  |     |  |     +--rw dscp?              uint8
            |  |  |     |  |     +--rw tos?               uint8
            |  |  |     |  |     +--rw dot1p?             uint8
            |  |  |     |  |     +--rw ipv4-src-prefix?   inet:ipv4-prefix
            |  |  |     |  |     +--rw ipv6-src-prefix?   inet:ipv6-prefix
            |  |  |     |  |     +--rw ipv4-dst-prefix?   inet:ipv4-prefix
            |  |  |     |  |     +--rw ipv6-dst-prefix?   inet:ipv6-prefix
            |  |  |     |  |     +--rw l4-src-port?       uint16
            |  |  |     |  |     +--rw l4-dst-port?       uint16
            |  |  |     |  |     +--rw protocol-field?    union
            |  |  |     |  +--:(match-application)
            |  |  |     |     +--rw match-application?   identityref
            |  |  |     +--rw target-class-id?     string
            |  |  +--rw qos-profile
            |  |     +--rw (qos-profile)?
            |  |        +--:(standard)
            |  |        |  +--rw profile?   string
            |  |        +--:(custom)
            |  |           +--rw classes {qos-custom}?
            |  |              +--rw class* [class-id]
            |  |                 +--rw class-id                 string
            |  |                 +--rw rate-limit?              uint8
            |  |                 +--rw priority-level?          uint8
            |  |                 +--rw guaranteed-bw-percent?   uint8
            |  +--rw carrierscarrier {carrierscarrier}?
            |  |  +--rw signalling-type?   enumeration
            |  +--rw multicast {multicast}?
            |     +--rw multicast-site-type?            enumeration
            |     +--rw multicast-transport-protocol
            |     |  +--rw ipv4?   boolean {ipv4}?
            |     |  +--rw ipv6?   boolean {ipv6}?
            |     +--rw protocol-type?                  enumeration
            +--rw traffic-protection {fast-reroute}?
            |  +--rw enabled?   boolean
            +--rw routing-protocols
            |  +--rw routing-protocol* [type]
            |     +--rw type      identityref
            |     +--rw ospf {rtg-ospf}?
            |     |  +--rw address-family*   identityref
            |     |  +--rw area-address?     yang:dotted-quad
            |     |  +--rw metric?           uint16
            |     |  +--rw sham-links {rtg-ospf-sham-link}?
            |     |     +--rw sham-link* [target-site]
            |     |        +--rw target-site    svc-id
```

```
            |     |        +--rw metric?        uint16
            |     +--rw bgp {rtg-bgp}?
            |     |  +--rw autonomous-system?   uint32
            |     |  +--rw address-family*      identityref
            |     +--rw static
            |     |  +--rw cascaded-lan-prefixes
            |     |     +--rw ipv4-lan-prefixes* [lan next-hop] {ipv4}?
            |     |     |  +--rw lan         inet:ipv4-prefix
            |     |     |  +--rw lan-tag?    string
            |     |     |  +--rw next-hop    inet:ipv4-address
            |     |     +--rw ipv6-lan-prefixes* [lan next-hop] {ipv6}?
            |     |        +--rw lan         inet:ipv6-prefix
            |     |        +--rw lan-tag?    string
            |     |        +--rw next-hop    inet:ipv6-address
            |     +--rw rip {rtg-rip}?
            |     |  +--rw address-family*   identityref
            |     +--rw vrrp {rtg-vrrp}?
            |        +--rw address-family*   identityref
            +--rw site-network-access
               +--rw access-diversity {site-diversity}?
               |  +--rw groups
               |  |  +--rw group* [group-id]
               |  |     +--rw group-id    string
               |  +--rw constraints
               |     +--rw constraint* [constraint-type]
               |        +--rw constraint-type    identityref
               |        +--rw target
               |           +--rw (target-flavor)?
               |              +--:(id)
               |              |  +--rw group* [group-id]
               |              |     +--rw group-id    string
               |              +--:(all-accesses)
               |              |  +--rw all-other-accesses?   empty
               |              +--:(all-groups)
               |                 +--rw all-other-groups?     empty
               +--rw bearer
               |  +--rw requested-type {requested-type}?
               |  |  +--rw requested-type?   string
               |  |  +--rw strict?           boolean
               |  +--rw always-on?          boolean {always-on}?
               |  +--rw bearer-reference?   string {bearer-reference}?
               +--rw ip-connection
               |  +--rw ipv4 {ipv4}?
               |  |  +--rw address-allocation-type?   identityref
               |  |  +--rw addresses
               |  |     +--rw provider-address?   inet:ipv4-address
               |  |     +--rw customer-address?   inet:ipv4-address
               |  |     +--rw mask?               uint8
```

```
               |  +--rw ipv6 {ipv6}?
               |  |  +--rw address-allocation-type?   identityref
               |  |  +--rw addresses
               |  |     +--rw provider-address?   inet:ipv6-address
               |  |     +--rw customer-address?   inet:ipv6-address
               |  |     +--rw mask?               uint8
               |  +--rw oam
               |     +--rw bfd {bfd}?
               |        +--rw bfd-enabled?    boolean
               |        +--rw (holdtime)?
               |           +--:(profile)
               |           |  +--rw profile-name?   string
               |           +--:(fixed)
               |              +--rw fixed-value?    uint32
               +--rw security
               |  +--rw authentication
               |  +--rw encryption {encryption}?
               |     +--rw enabled?              boolean
               |     +--rw layer?                enumeration
               |     +--rw encryption-profile
               |        +--rw (profile)?
               |           +--:(provider-profile)
               |           |  +--rw profile-name?    string
               |           +--:(customer-profile)
               |              +--rw algorithm?       string
               |              +--rw (key-type)?
               |                 +--:(psk)
               |                 |  +--rw preshared-key?   string
               |                 +--:(pki)
               +--rw service
               |  +--rw svc-input-bandwidth?    uint32
               |  +--rw svc-output-bandwidth?   uint32
               |  +--rw svc-mtu?                uint16
               |  +--rw qos {qos}?
               |  |  +--rw qos-classification-policy
               |  |  |  +--rw rule* [id]
               |  |  |     +--rw id                  uint16
               |  |  |     +--rw (match-type)?
               |  |  |     |  +--:(match-flow)
               |  |  |     |  |  +--rw match-flow
               |  |  |     |  |     +--rw dscp?              uint8
               |  |  |     |  |     +--rw tos?               uint8
               |  |  |     |  |     +--rw dot1p?             uint8
               |  |  |     |  |     +--rw ipv4-src-prefix?   inet:ipv4-prefix
               |  |  |     |  |     +--rw ipv6-src-prefix?   inet:ipv6-prefix
               |  |  |     |  |     +--rw ipv4-dst-prefix?   inet:ipv4-prefix
               |  |  |     |  |     +--rw ipv6-dst-prefix?   inet:ipv6-prefix
               |  |  |     |  |     +--rw l4-src-port?       uint16
```

```
|  |  |     |  |     +--rw l4-dst-port?       uint16
|  |  |     |  |     +--rw protocol-field?    union
|  |  |     |  +--:(match-application)
|  |  |     |     +--rw match-application?   identityref
|  |  |     +--rw target-class-id?     string
|  |  +--rw qos-profile
|  |     +--rw (qos-profile)?
|  |        +--:(standard)
|  |        |  +--rw profile?   string
|  |        +--:(custom)
|  |           +--rw classes {qos-custom}?
|  |              +--rw class* [class-id]
|  |                 +--rw class-id                 string
|  |                 +--rw rate-limit?              uint8
|  |                 +--rw priority-level?          uint8
|  |                 +--rw guaranteed-bw-percent?   uint8
|  +--rw carrierscarrier {carrierscarrier}?
|  |  +--rw signalling-type?   enumeration
|  +--rw multicast {multicast}?
|     +--rw multicast-site-type?            enumeration
|     +--rw multicast-transport-protocol
|     |  +--rw ipv4?   boolean {ipv4}?
|     |  +--rw ipv6?   boolean {ipv6}?
|     +--rw protocol-type?                  enumeration
+--rw routing-protocols
|  +--rw routing-protocol* [type]
|     +--rw type      identityref
|     +--rw ospf {rtg-ospf}?
|     |  +--rw address-family*   identityref
|     |  +--rw area-address?     yang:dotted-quad
|     |  +--rw metric?           uint16
|     |  +--rw sham-links {rtg-ospf-sham-link}?
|     |     +--rw sham-link* [target-site]
|     |        +--rw target-site    svc-id
|     |        +--rw metric?        uint16
|     +--rw bgp {rtg-bgp}?
|     |  +--rw autonomous-system?   uint32
|     |  +--rw address-family*      identityref
|     +--rw static
|     |  +--rw cascaded-lan-prefixes
|     |     +--rw ipv4-lan-prefixes* [lan next-hop] {ipv4}?
|     |     |  +--rw lan         inet:ipv4-prefix
|     |     |  +--rw lan-tag?    string
|     |     |  +--rw next-hop    inet:ipv4-address
|     |     +--rw ipv6-lan-prefixes* [lan next-hop] {ipv6}?
|     |        +--rw lan         inet:ipv6-prefix
|     |        +--rw lan-tag?    string
|     |        +--rw next-hop    inet:ipv6-address
```

```
               |     +--rw rip {rtg-rip}?
               |     |  +--rw address-family*   identityref
               |     +--rw vrrp {rtg-vrrp}?
               |        +--rw address-family*   identityref
               +--rw availability
               |  +--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 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.

```
                                    Management system
                     <------------------------------------------------->
                                                 Request RT
                      +-----------------------+  Topo a2a   +----------+
           RestConf   |                       |  ----->     |          |
   User ------------- | Service Orchestration |             |NetworkOSS|
           l3vpn-svc  |                       |  <-----     |          |
             model    +-----------------------+   Response  +----------+
                                                  RT1,RT2
```

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.

```
                      +----------------------------+
           RestConf   |                            |
   User ------------- |   Service Orchestration    |
           l3vpn-svc  |                            |
            model     |                            |
                      |  RT pool : 10:1->10:10000  |
                      |  RT pool : 20:50->20:5000  |
                      +----------------------------+
```

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

```
   +------------------------------------------------------------+
   |  VPN1_Site1 ------ PE1               PE2 ------ VPN1_Site2 |
   |                                                            |
   |  VPN1_Site3 ------ PE3               PE4 ------ VPN1_Site4 |
   +------------------------------------------------------------+
```

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

```
   +-------------------------------------------------------------+
   |   Hub_Site1 ------ PE1               PE2 ------ Spoke_Site1 |
   |                          +----------------------------------+
   |                          |
   |                          +----------------------------------+
   |   Hub_Site2 ------ PE3               PE4 ------ Spoke_Site2 |
   +-------------------------------------------------------------+
```

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.

```
      Hub_Site ------- (VRF_Hub)  PE1
                                 (VRF_Spoke)
                                   /  |
   Spoke_Site1 -------------------+   |
                                      |
   Spoke_Site2 -----------------------+
```

5.1.1.4. Hub and Spoke disjoint

```
   +-------------------------------------------------------------+
   |   Hub_Site1 ------ PE1               PE2 ------ Spoke_Site1 |
   +--------------------------+  +-------------------------------+
                              |  |
   +--------------------------+  +-------------------------------+
   |   Hub_Site2 ------ PE3               PE4 ------ Spoke_Site2 |
   +-------------------------------------------------------------+
```

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.

Hub and spoke disjoint can also be seen as two hub and spoke VPNs sharing with a common set of spoke sites.

5.1.2. Cloud access

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

Private cloud access may be addressed through NNIs as described in Section 5.14.

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 except" and "accept all except" authorization.

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

The "accept all except" 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 except", 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.

```
                  IPVPN
    ++++++++++++++++++++++++++++++++     +++++++++++
    +             Site 3           + --- +  Cloud1 +
    + Site 1                       +     +++++++++++
    +                              +
    + Site 2                       + --- ++++++++++++
    +                              +     + Internet +
    +            Site 4            +     ++++++++++++
    ++++++++++++++++++++++++++++++++
                 |
             ++++++++++
             + Cloud2 +
             ++++++++++
```

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-accesses>
     <cloud-access>
        <cloud-identifier>51</cloud-identifier>
        <nat-enabled>true</nat-enabled>
     </cloud-access>
    </cloud-accesses>
</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-accesses>
     <cloud-access>
        <cloud-identifier>1111111</cloud-identifier>
        <authorized-sites>
         <authorized-site>
            <site-id>site1</site-id>
            <site-id>site2</site-id>
         <authorized-site>
        </authorized-sites>
     <cloud-access>
    </cloud-accesses>
</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-accesses>
     <cloud-access>
        <cloud-identifier>22222222</cloud-identifier>
        <denied-sites>
         <denied-site>
            <site-id>site1</site-id>
         </denied-site>
        </denied-sites>
     </cloud-access>
    </cloud-accesses>
</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.

```
                       (SSM tree)
 Recv (IGMPv3) -- Site2 ------- PE2
                                           PE1 --- Site1 --- Source1
                                                         \
                                                          -- Source2

                        (ASM tree)
 Recv (IGMPv2) -- Site3 ------- PE3

                        (SSM tree)
 Recv (IGMPv3) -- Site4 ------- PE4
                                /
 Recv (IGMPv2) -- Site5 --------
                        (ASM tree)
```

In case of ASM flavor requested, this model requires to fill the rp and rp-discovery parameters. Multiple RP to group mappings can be created using the rp-group-mappings container. For each mapping, 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, user can request for rendez-vous point redundancy and/or optimal traffic delivery. Those parameters will help the service provider to select the appropriate technology to fulfill the customer service requirement : for instance, in case of request of optimal traffic delivery, service provider may decide to use Anycast-RP or RP-tree to SPT switchover.

In case of customer managed RP, the RP address must be filled in the RP to group mappings using the "rp-address" leaf. This leaf is not needed for provider managed RP.

User can define a specific rp-discovery mechanism like : auto-rp, static-rp, bsr-rp modes. By default, model considers static-rp if ASM is requested. A single rp-discovery mechanism is allowed for the VPN. "rp-discovery" can be used for provider and customer managed RPs. In case of provider managed RP, if the user wants to use bsr-rp as discovery protocol, service provider will consider the provider managed rp-group-mappings for bsr-rp. The service provider will so configure its selected RPs to be bsr-rp-candidates. In case of customer managed RP and bsr-rp discovery mechanism, the rp-address provided will be considered as bsr-rp candidate.

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.

```
            +-----------+             +-----------+
           /             \           /             \
SiteA --  |   VPN A       |   ---   |    VPN B      | --- SiteB
           \             /           \             / (Shared
            +-----------+             +-----------+   resources)
```

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 5.4.2.2. 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 under VPN A with an entry corresponding to VPN B and there is no service configuration requirement on VPN B.

Readers should note that even if there is no configuration requirement on VPN B, if VPN A lists VPN B as extranet, all sites in VPN B will gain access to all sites in VPN 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 5.3. Based on this, the requirements described in Section 5.3 regarding the site-role leaf are also applicable here.

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 5.4.2.

5.2. Site overview

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

```
                                                    +-------------+
                                                   /               \
     +------------------+                   +-----|      VPN1       |
     |                  |                   |      \               /
     |  New York Office | ----- (site) -----+       +-------------+
     |                  |                   |       +-------------+
     +------------------+                   |      /               \
                                            +-----|      VPN2       |
                                                   \               /
                                                    +-------------+
```

A site is composed of some characteristics :

- 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.

- 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.

```
     +------------------+             Site
     |                  |-----------------------------------
     |                  |****** (site-network-access#1) ******
     |  New York Office |
     |                  |****** (site-network-access#2) ******
     |                  |-----------------------------------
     +------------------+
```

Multiple site-network-accesses are used for instance in case of multihoming. Some other topology cases may also involve multiple site-network-accesses.

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 network accesses

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

- bearer : defines requirements of the attachment (below Layer 3).

- connection : defines Layer 3 protocol parameters of the attachment.

- availability : defines the site availability policy. Availability is defined in Section 5.6

Some parameters from the site can be configured also at the site-network-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. Those parameters will be described later in the document.

5.2.1.1. Bearer

Bearer defines the requirements for the site attachment to the provider network that are below Layer 3.

The bearer parameters will help to decide the access media to be used. This is further described in Section 5.5.2.

5.2.1.2. 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.

5.2.1.2.1. 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.

In the dynamic addressing mechanism, it is expected from service provider to provide at least the IP address, mask and default gateway information.

5.2.1.2.2. OAM

A customer may require a specific IP connectivity fault detection mechanism on the IP connection. The model supports BFD as mechanism proposed to the customer. This can be extended with other mechanisms by augmentation. The provider can propose some profiles to the customer depending of the service level the customer wants to achieve. Profile names must be communicated to the customer. This communication is out of scope of this document. Some fixed values for the holdtime period may also be imposed by the customer if the provider enables it.

5.2.1.3. Inheritance of parameters between site and site-network-access

Some parameters are available both at site level and site-network-access level. Defining a parameter at site level will provide inheritance to all site-network-accesses under the site. If a site-network-access has a parameter configured that is already defined at site level, the site-network-access parameter value will replace the site parameter value.

5.3. 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.

5.4. Site belonging to multiple VPNs

5.4.1. 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).

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

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

```
                                                      +--------+
   +------------------+             Site             /          \
   |                  |-----------------------------|            |
   |                  |***(site-network-access#1)***|    VPN1    |
   |  New York Office |                             |            |
   |                  |***(site-network-access#2)***|            |
   |                  |-----------------------------|            |
   +------------------+                              \          /
                                                      +--------+
```

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

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

```
                                                           +---------+
                                                      +---/----+      \
   +------------------+             Site             /   |      \      |
   |                  |--------------------------------- |       |VPNB |
   |                  |***(site-network-access#1)******* |       |     |
   |  New York Office |                             |    |       |     |
   |                  |***(site-network-access#2)*******  \      |    /
   |                  |-----------------------------| VPNA +-----|---+
   +------------------+                              \          /
                                                      +--------+
```

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.

Reaching VPN A or VPN B from New York office will be based on destination based routing. Having the same destination reachable from the two VPNs may cause routing troubles. This would be the role of the customer administration to ensure the appropriate mapping of its prefixes in each VPN.

5.4.1.3. 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.

```
   +------------------+         Site                      +--------+
   |                  |----------------------------------/          \
   |                  |****(site-network-access#1)******|    VPNB    |
   |  New York Office |                                  \          /
   |                  |                                   +--------+
   |                  |                                   +--------+
   |                  |                                  /          \
   |                  |****(site-network-access#2)******|    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.

```
  +------------------+         Site                      +-----+
  |                  |----------------------------------/      +----+
  |                  |****(site-network-access#1)******| VPNB /       \
  |  New York Office |                                  \     | VPN C  |
  |                  |                                   +----\       /
  |                  |                                         +-----+
  |                  |
  |                  |                                   +------+
  |                  |                                  /       +-----+
  |                  |****(site-network-access#2)******|  VPNA /       \
  |                  |                                  \      | VPN D |
  |                  |                                   +------\     /
  |                  |-----------------------------------        +---+
  +------------------+
```

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.

```
  +------------------+         Site                      +-----+
  |                  |----------------------------------/      +----+
  |                  |****(site-network-access#1)******| VPNB /       \
  |  New York Office |****(site-network-access#2)******\     | VPN C  |
  |                  |                                   +----\       /
  |                  |                                         +-----+
  |                  |
  |                  |                                   +------+
  |                  |                                  /       +-----+
  |                  |****(site-network-access#3)******|  VPNA /       \
  |                  |****(site-network-access#4)****** \      | VPN D |
  |                  |                                   +------\     /
  |                  |-----------------------------------        +---+
  +------------------+
```

5.4.2. 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.

- Reference a VPN policy for more complex attachments.

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

5.4.2.1. 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.

5.4.2.2. 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.

```
   +--------------------------------------------------------------+
   |  VPN2_Site3 ------ PE7                                       |
   +-------------------------+                                    |
                             |                                    |
   +-------------------------+                                    |
   |  VPN2_Site1 ------ PE3               PE4 ------ VPN2_Site2   |
   +----------------------------------+                           |
                                      |                           |
   +------------------------------------------------------------+ |
   |  VPN3_Site1 ------ PE5           |   PE6 ------ VPN3_Site2 | |
   +------------------------------------------------------------+ |
                                      |                           |
                                      +---------------------------+
```

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>
```

5.5. Deciding where to connect the site

The management system will have to decide where to connect each site-network-access of a particular site to the provider network (PE, aggregation switch ...).

The current model proposes parameters and constraints that will help the management system to decide where to attach the site-network-access.

The management system SHOULD honor the customer constraints, if the constraint cannot be filled, the management system MUST not provision the site and SHOULD provide an information to the user. How the information is provided is out of scope of the document. It would then be up to the user to relax the constraint or not.

Parameters are just hints for management system for service placement.

In addition to parameters and constraints : the management system decision MAY be based on any other internal constraint that are up to the service provider : least load, distance ...

5.5.1. Parameter : 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.

```
                                        PoP#1 (New York)
                                     +---------+
                                     |   PE1   |
                Site #1 ---...       |   PE2   |
               (Atlantic City)       |   PE3   |
                                     +---------+

                                        PoP#2 (Washington)
                                     +---------+
                                     |   PE4   |
                                     |   PE5   |
                                     |   PE6   |
                                     +---------+

                                        PoP#3 (Philadelphia)
                                     +---------+
                                     |   PE7   |
                Site #2 ---...       |   PE2   |
               (Reston)              |   PE9   |
                                     +---------+
```

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.

5.5.2. Constraint/parameter : access type

The management system will need to elect the access method to connect the site to the customer (for example : PPP over ISDN, xSDL, leased line, Ethernet backhaul ...). The customer may provide some parameters/constraints that will provide hints to the management system.

The bearer container information SHOULD be used as first input :

- The "requested-type" provides an information about the media type the customer would like. If the "strict" leaf is equal to "true", this MUST be considered as a strict constraint, so the management system cannot connect the site with another media type. If the "strict" leaf is equal to "false" (default), if the requested-type cannot be fullfilled, the management system can select another type. The supported media types SHOULD be communicated by the service provider to the customer by a mechanism that is out of scope of the document.

- The "always-on" leaf defines a strict constraint : if set to "true", the management system MUST elect a media type which is always-on (this means no Dial access type).

- The "bearer-reference" is used in case the customer has already ordered a network connection to the service provider apart of the IPVPN site and wants to reuse this connection. The string used in an internal reference from the service provider describing the already available connection. This is also a strict requirement that cannot be relaxed. How the reference is given to the customer is out of scope of the document but as a pure example, when the customer ordered the bearer (through a process out of this model), the service provider may had provided the bearer reference that can be used for provisionning services on top.

Other parameters like the requested svc-input-bandwidth, svc-output-bandwidth MAY help to decide the access type to be used. Any other internal parameters from the service provider can be used in addition.

5.5.3. Constraint : access diversity

Each site-network-access may have one or more constraints that would drive the placement of the access.

In order to help the different placement scenarios, a site-network-access may be tagged using one or multiple group identifiers. The group identifier is a string so can accomodate both explicit naming of a group of sites (e.g. "multi-homed-set1" or "subvpn") or using a numbered id (e.g. 12345678). The meaning of each group-id is local to each customer administrator. And the management system MUST ensure that different customers can use the same group-ids. One or more group-id can also be defined at site-level, as a consequence, all site-network-accesses under the site MUST inherit the group-ids of the site they are belonging to. When, in addition to the site group-ids, some group-ids are defined at site-network-access level, the management system MUST consider the union of all groups (site level and site network access level) for this particular site network access.

For a particular currently configured site-network-access, each constraint MUST be expressed against a targeted set of site-network-accesses, the currently configured site-network-access MUST never be taken into account in the targeted set : e.g. "I want my current site-network-access to be not be connected on the same PoP as the site-network-accesses that are part of group 10". The set of site-network-accesses against which the constraint is evaluated can be expressed as a list of groups or all-other-accesses or all-other-groups. "all-other-accesses" means that the current site-network-access constraint MUST be evaluated against all the other site-network-accesses belonging to the current site. "all-other-groups" means that the constraint MUST be evaluated against all groups the current site-network-access is not belonging to.

The current model proposes multiple constraint-types :

pe-diverse : the current site-network-access MUST not be connected to the same PE as the targeted site-network-accesses.

pop-diverse : the current site-network-access MUST not be connected to the same PoP as the targeted site-network-accesses.

linecard-diverse : the current site-network-access MUST not be connected to the same linecard as the targeted site-network-accesses.

same-pe : the current site-network-access MUST be connected to the same PE as the targeted site-network-accesses.

same-bearer : the current site-network-access MUST be connected using the same bearer as the targeted site-network-accesses.

Those constraint-types could be extended through augmentation.

Each constraint is expressed as "I want my current site-network-access to be <constraint-type> (e.g. pe-diverse, pop-diverse) from those <target> site-network-accesses". In addition,

The group-id used to target some site-network-accesses may be the same as the one used by the current site-network-access. This ease configuration of scenarios where a group of site-network-access has a constraint between each other. As an example if we want a set of sites (site#1 up to #5) to be all connected on a different PE, we can tag them with the same group-id and express a pe-diverse constraint for this group-id.

```
<site>
 <site-id>SITE1</site-id>
 <site-network-accesses>
  <site-network-access>
   <site-network-access-id>1</site-network-access-id>
   <access-diversity>
    <groups>
     <group>
      <group-id>10</group-id>
     </group>
    </groups>
    <constraints>
     <constraint>
      <constraint-type>pe-diverse</constraint-type>
      <target>
       <group>
        <group-id>10</group-id>
       </group>
      </target>
     </constraint>
    </constraints>
   </access-diversity>
   <vpn-attachment>
    <vpn-id>VPNA</vpn-id>
    <site-role>spoke-role</site-role>
   </vpn-attachment>
  </site-network-access>
 </site-network-accesses>
</site>
<site>
 <site-id>SITE2</site-id>
 <site-network-accesses>
  <site-network-access>
   <site-network-access-id>1</site-network-access-id>
   <access-diversity>
    <groups>
```

```
        <group>
         <group-id>10</group-id>
        </group>
       </groups>
       <constraints>
        <constraint>
         <constraint-type>pe-diverse</constraint-type>
         <target>
          <group>
           <group-id>10</group-id>
          </group>
         </target>
        </constraint>
       </constraints>
      </access-diversity>
      <vpn-attachment>
       <vpn-id>VPNA</vpn-id>
       <site-role>spoke-role</site-role>
      </vpn-attachment>
     </site-network-access>
    </site-network-accesses>
   </site>
   ...
   <site>
    <site-id>SITE5</site-id>
    <site-network-accesses>
     <site-network-access>
      <site-network-access-id>1</site-network-access-id>
      <access-diversity>
       <groups>
        <group>
         <group-id>10</group-id>
        </group>
       </groups>
       <constraints>
        <constraint>
         <constraint-type>pe-diverse</constraint-type>
         <target>
          <group>
           <group-id>10</group-id>
          </group>
         </target>
        </constraint>
       </constraints>
      </access-diversity>
      <vpn-attachment>
       <vpn-id>VPNA</vpn-id>
       <site-role>spoke-role</site-role>
```

```
   </vpn-attachment>
  </site-network-access>
 </site-network-accesses>
</site>
```

The group-id used to target some site-network-accesses may be also different as the one used by the current site-network-access. This is used to express that a group of site has some constraint against another group of sites, but there may not be constraint inside the group itself. As an example, if we consider a set of 6 sites with two sets and we want to ensure that a site in the first set must be pop-diverse from a site in the second set.

```
<site>
 <site-id>SITE1</site-id>
 <site-network-accesses>
  <site-network-access>
   <site-network-access-id>1</site-network-access-id>
   <access-diversity>
    <groups>
     <group>
      <group-id>10</group-id>
     </group>
    </groups>
    <constraints>
     <constraint>
      <constraint-type>pop-diverse</constraint-type>
      <target>
       <group>
        <group-id>20</group-id>
       </group>
      </target>
     </constraint>
    </constraints>
   </access-diversity>
   <vpn-attachment>
    <vpn-id>VPNA</vpn-id>
    <site-role>spoke-role</site-role>
   </vpn-attachment>
  </site-network-access>
 </site-network-accesses>
</site>
<site>
 <site-id>SITE2</site-id>
 <site-network-accesses>
  <site-network-access>
   <site-network-access-id>1</site-network-access-id>
   <access-diversity>
```

```
       <groups>
        <group>
         <group-id>10</group-id>
        </group>
       </groups>
       <constraints>
        <constraint>
         <constraint-type>pop-diverse</constraint-type>
         <target>
          <group>
           <group-id>20</group-id>
          </group>
         </target>
        </constraint>
       </constraints>
      </access-diversity>
      <vpn-attachment>
       <vpn-id>VPNA</vpn-id>
       <site-role>spoke-role</site-role>
      </vpn-attachment>
     </site-network-access>
    </site-network-accesses>
   </site>
   ...
   <site>
    <site-id>SITE5</site-id>
    <site-network-accesses>
     <site-network-access>
      <site-network-access-id>1</site-network-access-id>
      <access-diversity>
       <groups>
        <group>
         <group-id>20</group-id>
        </group>
       </groups>
       <constraints>
        <constraint>
         <constraint-type>pop-diverse</constraint-type>
         <target>
          <group>
           <group-id>10</group-id>
          </group>
         </target>
        </constraint>
       </constraints>
      </access-diversity>
      <vpn-attachment>
       <vpn-id>VPNA</vpn-id>
```

```
       <site-role>spoke-role</site-role>
      </vpn-attachment>
     </site-network-access>
    </site-network-accesses>
   </site>
   <site>
    <site-id>SITE6</site-id>
    <site-network-accesses>
     <site-network-access>
      <site-network-access-id>1</site-network-access-id>
      <access-diversity>
       <groups>
        <group>
         <group-id>20</group-id>
        </group>
       </groups>
       <constraints>
        <constraint>
         <constraint-type>pop-diverse</constraint-type>
         <target>
          <group>
           <group-id>10</group-id>
          </group>
         </target>
        </constraint>
       </constraints>
      </access-diversity>
      <vpn-attachment>
       <vpn-id>VPNA</vpn-id>
       <site-role>spoke-role</site-role>
      </vpn-attachment>
     </site-network-access>
    </site-network-accesses>
   </site>
```

5.5.4. Impossible access placement

Some impossible placement scenarios may be created through the proposed configuration framework. Impossible scenarios could be too restrictive constraints leading to impossible placement in the network or conflicting constraints that would also lead to impossible placement. An example of conflicting rules would be to ask a site-network-access#1 to be pe-diverse from a site-network-access#2 and to ask at the same time that site-network-access#2 to be on the same PE as site-network-access#1. When the management system cannot place the access, it SHOULD return an error message indicating that placement was not possible.

5.5.5. Examples of access placement

5.5.5.1. Multihoming

The customer wants to create a multihomed site. The site will be composed of two site-network-accesses and the customer wants the two site-network-accesses to be meshed on different PoPs for resiliency purpose.

```
                                        PoP#1
                                     +---------+
   |                                 |   PE1   |
   |---site_network_access#1 ----    |   PE2   |
   |                                 |   PE3   |
   |                                 +---------+
   |
   |                                    PoP#2
   |                                 +---------+
   |                                 |   PE4   |
   |---site_network_access#2 ----    |   PE5   |
   |                                 |   PE6   |
   |                                 +---------+
```

This scenario could be expressed in the following way :

```
   <site>
    <site-id>SITE1</site-id>
    <site-network-accesses>
     <site-network-access>
      <site-network-access-id>1</site-network-access-id>
      <access-diversity>
       <groups>
        <group>
         <group-id>10</group-id>
        </group>
       </groups>
       <constraints>
        <constraint>
         <constraint-type>pop-diverse</constraint-type>
         <target>
          <group>
           <group-id>20</group-id>
          </group>
         </target>
        </constraint>
       </constraints>
      </access-diversity>
```

```
      <vpn-attachment>
       <vpn-id>VPNA</vpn-id>
       <site-role>spoke-role</site-role>
      </vpn-attachment>
     </site-network-access>
     <site-network-access>
      <site-network-access-id>2</site-network-access-id>
      <access-diversity>
       <groups>
        <group>
         <group-id>20</group-id>
        </group>
       </groups>
       <constraints>
        <constraint>
         <constraint-type>pop-diverse</constraint-type>
         <target>
          <group>
           <group-id>10</group-id>
          </group>
         </target>
        </constraint>
       </constraints>
      </access-diversity>
      <vpn-attachment>
       <vpn-id>VPNA</vpn-id>
       <site-role>spoke-role</site-role>
      </vpn-attachment>
     </site-network-access>
    </site-network-accesses>
   </site>
```

But it can also be expressed as :

```
   <site>
    <site-id>SITE1</site-id>
    <site-network-accesses>
     <site-network-access>
      <site-network-access-id>1</site-network-access-id>
      <access-diversity>
       <constraints>
        <constraint>
         <constraint-type>pop-diverse</constraint-type>
         <target>
          <all-other-accesses/>
         </target>
        </constraint>
       </constraints>
      </access-diversity>
      <vpn-attachment>
       <vpn-id>VPNA</vpn-id>
       <site-role>spoke-role</site-role>
      </vpn-attachment>
     </site-network-access>
     <site-network-access>
      <site-network-access-id>2</site-network-access-id>
      <access-diversity>
       <constraints>
        <constraint>
         <constraint-type>pop-diverse</constraint-type>
         <target>
          <all-other-accesses/>
         </target>
        </constraint>
       </constraints>
      </access-diversity>
      <vpn-attachment>
       <vpn-id>VPNA</vpn-id>
       <site-role>spoke-role</site-role>
      </vpn-attachment>
     </site-network-access>
    </site-network-accesses>
   </site>
```

5.5.5.2. Site offload

The customer has a 6 branch offices in a particular region and he wants to prevent to have all branch offices on the same PE.

He wants to express that 3 branch offices cannot be connected not on the same linecard. But the other branch offices must be connected on

a different PoP. Those other branch offices cannot also be connected on the same linecard.

```
                                     PoP#1
                                  +---------+
                                  |   PE1   |
            Office#1 ---...       |   PE2   |
            Office#2 ---...       |   PE3   |
            Office#3 ---...       |   PE4   |
                                  +---------+


                                     PoP#2
                                  +---------+
            Office#4 ---...       |   PE4   |
            Office#5 ---...       |   PE5   |
            Office#6 ---...       |   PE6   |
                                  +---------+
```

This scenario could be expressed in the following way :

- We need to create two sets of sites : set#1 composed of Office#1 up to 3, set#2 composed of Office#4 up to 6.

- Sites within set#1 must be pop-diverse from sites within set#2 and vice versa.

- Sites within set#1 must be linecard-diverse from other sites in set#1 (same for set#2).

```
<site>
 <site-id>SITE1</site-id>
 <site-network-accesses>
  <site-network-access>
   <site-network-access-id>1</site-network-access-id>
   <access-diversity>
    <groups>
     <group>
      <group-id>10</group-id>
     </group>
    </groups>
    <constraints>
     <constraint>
      <constraint-type>pop-diverse</constraint-type>
      <target>
       <group>
```

```
              <group-id>20</group-id>
             </group>
            </target>
           </constraint>
           <constraint>
            <constraint-type>linecard-diverse</constraint-type>
            <target>
             <group>
              <group-id>10</group-id>
             </group>
            </target>
           </constraint>
          </constraints>
         </access-diversity>
         <vpn-attachment>
          <vpn-id>VPNA</vpn-id>
          <site-role>spoke-role</site-role>
         </vpn-attachment>
        </site-network-access>
      </site>
      <site>
       <site-id>SITE2</site-id>
       <site-network-accesses>
        <site-network-access>
         <site-network-access-id>1</site-network-access-id>
         <access-diversity>
          <groups>
           <group>
            <group-id>10</group-id>
           </group>
          </groups>
          <constraints>
           <constraint>
            <constraint-type>pop-diverse</constraint-type>
            <target>
             <group>
              <group-id>20</group-id>
             </group>
            </target>
           </constraint>
           <constraint>
            <constraint-type>linecard-diverse</constraint-type>
            <target>
             <group>
              <group-id>10</group-id>
             </group>
            </target>
           </constraint>
```

```
       </constraints>
      </access-diversity>
      <vpn-attachment>
       <vpn-id>VPNA</vpn-id>
       <site-role>spoke-role</site-role>
      </vpn-attachment>
     </site-network-access>
   </site>
   <site>
    <site-id>SITE3</site-id>
    <site-network-accesses>
     <site-network-access>
      <site-network-access-id>1</site-network-access-id>
      <access-diversity>
       <groups>
        <group>
         <group-id>10</group-id>
        </group>
       </groups>
       <constraints>
        <constraint>
         <constraint-type>pop-diverse</constraint-type>
         <target>
          <group>
           <group-id>20</group-id>
          </group>
         </target>
        </constraint>
        <constraint>
         <constraint-type>linecard-diverse</constraint-type>
         <target>
          <group>
           <group-id>10</group-id>
          </group>
         </target>
        </constraint>
       </constraints>
      </access-diversity>
      <vpn-attachment>
       <vpn-id>VPNA</vpn-id>
       <site-role>spoke-role</site-role>
      </vpn-attachment>
     </site-network-access>
    </site-network-accesses>
   </site>

   <site>
    <site-id>SITE4</site-id>
```

```
       <site-network-accesses>
        <site-network-access>
         <site-network-access-id>1</site-network-access-id>
         <access-diversity>
          <groups>
           <group>
            <group-id>20</group-id>
           </group>
          </groups>
          <constraints>
           <constraint>
            <constraint-type>pop-diverse</constraint-type>
            <target>
             <group>
              <group-id>10</group-id>
             </group>
            </target>
           </constraint>
           <constraint>
            <constraint-type>linecard-diverse</constraint-type>
            <target>
             <group>
              <group-id>20</group-id>
             </group>
            </target>
           </constraint>
          </constraints>
         </access-diversity>
         <vpn-attachment>
          <vpn-id>VPNA</vpn-id>
          <site-role>spoke-role</site-role>
         </vpn-attachment>
        </site-network-access>
      </site>
      <site>
       <site-id>SITE5</site-id>
       <site-network-accesses>
        <site-network-access>
         <site-network-access-id>1</site-network-access-id>
         <access-diversity>
          <groups>
           <group>
            <group-id>20</group-id>
           </group>
          </groups>
          <constraints>
           <constraint>
            <constraint-type>pop-diverse</constraint-type>
```

```
         <target>
          <group>
           <group-id>10</group-id>
          </group>
         </target>
        </constraint>
        <constraint>
         <constraint-type>linecard-diverse</constraint-type>
         <target>
          <group>
           <group-id>20</group-id>
          </group>
         </target>
        </constraint>
       </constraints>
      </access-diversity>
      <vpn-attachment>
       <vpn-id>VPNA</vpn-id>
       <site-role>spoke-role</site-role>
      </vpn-attachment>
     </site-network-access>
   </site>
   <site>
    <site-id>SITE6</site-id>
    <site-network-accesses>
     <site-network-access>
      <site-network-access-id>1</site-network-access-id>
      <access-diversity>
       <groups>
        <group>
         <group-id>20</group-id>
        </group>
       </groups>
       <constraints>
        <constraint>
         <constraint-type>pop-diverse</constraint-type>
         <target>
          <group>
           <group-id>10</group-id>
          </group>
         </target>
        </constraint>
        <constraint>
         <constraint-type>linecard-diverse</constraint-type>
         <target>
          <group>
           <group-id>20</group-id>
          </group>
```

```
         </target>
        </constraint>
       </constraints>
      </access-diversity>
      <vpn-attachment>
       <vpn-id>VPNA</vpn-id>
       <site-role>spoke-role</site-role>
      </vpn-attachment>
     </site-network-access>
    </site-network-accesses>
   </site>
```

5.5.5.3. Parallel links

To increase its site bandwidth at a cheaper cost, a customer wants to order to parallel site-network-accesses that will be connected to the same PE.

```
          *******SNA1**********
   Site 1 *******SNA2**********  PE1
```

This scenario could be expressed in the following way :

```
   <site>
    <site-id>SITE1</site-id>
    <site-network-accesses>
     <site-network-access>
      <site-network-access-id>1</site-network-access-id>
      <access-diversity>
       <groups>
        <group>
         <group-id>PE-linkgrp-1</group-id>
        </group>
       </groups>
       <constraints>
        <constraint>
         <constraint-type>same-pe</constraint-type>
         <target>
          <group>
           <group-id>PE-linkgrp-1</group-id>
          </group>
         </target>
        </constraint>
       </constraints>
      </access-diversity>
      <vpn-attachment>
```

```
       <vpn-id>VPNB</vpn-id>
       <site-role>spoke-role</site-role>
      </vpn-attachment>
     </site-network-access>
     <site-network-access>
      <site-network-access-id>2</site-network-access-id>
      <access-diversity>
       <groups>
        <group>
         <group-id>PE-linkgrp-1</group-id>
        </group>
       </groups>
       <constraints>
        <constraint>
         <constraint-type>same-pe</constraint-type>
         <target>
          <group>
           <group-id>PE-linkgrp-1</group-id>
          </group>
         </target>
        </constraint>
       </constraints>
      </access-diversity>
      <vpn-attachment>
       <vpn-id>VPNB</vpn-id>
       <site-role>spoke-role</site-role>
      </vpn-attachment>
     </site-network-access>
    </site-network-accesses>
   </site>
```

5.5.5.4. SubVPN with multihoming

A customer has site which is dual-homed, the dual-homing must be done on two different PEs. The customer wants also to implement two subVPNs on those multi-homed accesses.

```
  +------------------+         Site                      +------+
  |                  |----------------------------------/      +-----+
  |                  |****(site-network-access#1)******| VPNB /       \
  |  New York Office |****(site-network-access#2)*************| VPN C  |
  |                  |                                  +----\       /
  |                  |                                        +-----+
  |                  |
  |                  |                                  +------+
  |                  |                                 /       +-----+
  |                  |****(site-network-access#3)******|  VPNB /       \
  |                  |****(site-network-access#4)**************| VPN C |
  |                  |                                  +------\     /
  |                  |-----------------------------------        +---+
  +------------------+
```

This scenario could be expressed in the following way :

- The site will have 4 site network accesses (2 subVPN coupled with dual homing).

- Site-network-access#1 and #3 will correspond to the multihoming of the subVPN B. A PE-diverse constraint is required between them.

- Site-network-access#2 and #4 will correspond to the multihoming of the subVPN C. A PE-diverse constraint is required between them.

- To ensure proper usage of the same bearer for the subVPN, site-network-access #1 and #2 must share the same bearer as site-network-access #3 and #4.

```
<site>
 <site-id>SITE1</site-id>
 <site-network-accesses>
  <site-network-access>
   <site-network-access-id>1</site-network-access-id>
   <access-diversity>
    <groups>
     <group>
      <group-id>dual-homed-1</group-id>
     </group>
    </groups>
    <constraints>
     <constraint>
      <constraint-type>pe-diverse</constraint-type>
      <target>
       <group>
        <group-id>dual-homed-2</group-id>
       </group>
```

```
         </target>
        </constraint>
        <constraint>
         <constraint-type>same-bearer</constraint-type>
         <target>
          <group>
           <group-id>dual-homed-1</group-id>
          </group>
         </target>
        </constraint>
       </constraints>
      </access-diversity>
      <vpn-attachment>
       <vpn-id>VPNB</vpn-id>
       <site-role>spoke-role</site-role>
      </vpn-attachment>
     </site-network-access>
     <site-network-access>
      <site-network-access-id>2</site-network-access-id>
      <access-diversity>
       <groups>
        <group>
         <group-id>dual-homed-1</group-id>
        </group>
       </groups>
       <constraints>
        <constraint>
         <constraint-type>pe-diverse</constraint-type>
         <target>
          <group>
           <group-id>dual-homed-2</group-id>
          </group>
         </target>
        </constraint>
        <constraint>
         <constraint-type>same-bearer</constraint-type>
         <target>
          <group>
           <group-id>dual-homed-1</group-id>
          </group>
         </target>
        </constraint>
       </constraints>
      </access-diversity>
      <vpn-attachment>
       <vpn-id>VPNC</vpn-id>
       <site-role>spoke-role</site-role>
      </vpn-attachment>
```

```
     </site-network-access>
      <site-network-access-id>3</site-network-access-id>
      <access-diversity>
       <groups>
        <group>
         <group-id>dual-homed-2</group-id>
        </group>
       </groups>
       <constraints>
        <constraint>
         <constraint-type>pe-diverse</constraint-type>
         <target>
          <group>
           <group-id>dual-homed-1</group-id>
          </group>
         </target>
        </constraint>
        <constraint>
         <constraint-type>same-bearer</constraint-type>
         <target>
          <group>
           <group-id>dual-homed-2</group-id>
          </group>
         </target>
        </constraint>
       </constraints>
      </access-diversity>
      <vpn-attachment>
       <vpn-id>VPNB</vpn-id>
       <site-role>spoke-role</site-role>
      </vpn-attachment>
     </site-network-access>
     <site-network-access>
      <site-network-access-id>4</site-network-access-id>
      <access-diversity>
       <groups>
        <group>
         <group-id>dual-homed-2</group-id>
        </group>
       </groups>
       <constraints>
        <constraint>
         <constraint-type>pe-diverse</constraint-type>
         <target>
          <group>
           <group-id>dual-homed-1</group-id>
          </group>
         </target>
```

```
     </constraint>
     <constraint>
      <constraint-type>same-bearer</constraint-type>
      <target>
       <group>
        <group-id>dual-homed-2</group-id>
       </group>
      </target>
     </constraint>
    </constraints>
   </access-diversity>
   <vpn-attachment>
    <vpn-id>VPNC</vpn-id>
    <site-role>spoke-role</site-role>
   </vpn-attachment>
  </site-network-access>
 </site-network-accesses>
</site>
```

5.5.6. 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.

5.6. Site network access availability

A site may be multihomed, so having multiple site-network-accesses. Placement constraints defined in previous sections will help to ensure physical diversity.

When the site-network-accesses are placed on the network, a customer may want to use a particular routing policy on those accesses.

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 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.

```
Hub#1 LAN (Primary/backup)          Hub#2 LAN (Loadsharing)
  |                                                    |
  |     access-priority 1       access-priority 1    |
  |--- CE1 ------- PE1         PE3 --------- CE3 --- |
  |                                                    |
  |                                                    |
  |--- CE2 ------- PE2         PE4 --------- CE4 --- |
  |     access-priority 2       access-priority 1    |


                          PE5
                           |
                           |
                           |
                          CE5
                           |
                      Spoke#1 site (Single-homed)
```

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.

The access-priority has some limitation. A scenario like the previous one with 5 accesses but with the constraint of having traffic loadshared between A3 and A4 in case of A1 OR A2 being down is not achievable. But the authors consider that the access-priority covers most of the deployment use cases and the model can still be extended by augmentation to support new use cases.

5.7. Traffic protection

The service model supports the ability to protect traffic for the site. Protection provides a better availability to multihoming by, for example, using local-repair techniques in case of failures. The associated level of service guarantee would be based on an agreement between customer and service provider and is out of scope of this document.

```
       Site#1                            Site#2
   CE1 ----- PE1 -- P1            P3 -- PE3 ---- CE3
    |                              |              |
    |                              |              |
   CE2 ----- PE2 -- P2            P4 -- PE4 ---- CE4
             /
            /
   CE5 ----+
      Site#3
```

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.

5.8. Security

Security container defines customer specific security parameters for the site.

5.8.1. Authentication

The current model does not support any authentication parameters, but such parameters may be added in the authentication container through augmentation.

5.8.2. 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 profile or customer specific.

5.9. Management

The model proposes three types of common management options :

- provider-managed : the CE router is managed only by the provider. In this model, the responsibility boundary between SP and customer is between CE and customer network.

- customer-managed : the CE router is managed only by the customer. In this model, the responsibility boundary between SP and customer is between PE and CE.

- co-managed : the CE router is primarly managed by the provider and in addition SP lets customer accessing the CE for some configuration/monitoring purpose. In the co-managed mode the responsibility boundary is the same as the provider-managed model.

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

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

5.10. 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.

```
                               Rtg protocol
       192.0.2.0/24 ----- CE ----------------- PE1

                    Customer managed site

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

                    Provider managed site
```

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

5.10.1. 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</address-family>
        <address-family>ipv6</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</address-family>
    </rip>
  </routing-protocol>
  <routing-protocol>
    <type>ospf</type>
    <ospf>
        <address-family>ipv6</address-family>
    </ospf>
  </routing-protocol>
</routing-protocols>
```

5.10.2. 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
```

In this case, the customer has a default route to the service provider network.

5.10.3. 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
```

In this case, the customer has a default route to the service provider network.

5.10.4. 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
```

In this case, the customer has a default route to the service provider network.

5.10.5. 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
```

5.10.6. 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.

```
                             +---------------------+
                             |                     |
                     OSPF    |                     | OSPF
                     area 1  |                     | area 2
    (OSPF                    |                     |          (OSPF
    area 1) --- CE ---------- PE               PE ----- CE --- area 2)
                             |                     |
                             +---------------------+
```

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.

```
                           +------------------------+
                           |                        |
                           |                        |
                           |   PE (--shamlink--)PE  |
                           |    |                |  |
                           +----|----------------|--+
                                | OSPF area1     | OSPF area 1
                                |                |
                                CE1             CE2
                                |                |
                           (OSPF area1)       (OSPF area1)
                                |                |
                                +----------------+
```

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</address-family>
        <address-family>ipv6</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
 !
```

5.10.7. 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.

BGP activation requires SP to know the address of the customer peer. "static-address" allocation type for the IP connection MUST be used.

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

5.11. Service

The service defines service parameters associated with the site.

5.11.1. Bandwidth

The service bandwidth refers to the bandwidth requirement between PE and CE (WAN access bandwidth). The requested bandwidth is expressed as svc-input-bandwidth and svc-output-bandwidth in bit per seconds. Input/output direction is using customer site as reference : input bandwidth so means download bandwidth for the site, and output bandwidth means upload bandwidth for the site.

Using a different input and output bandwidth will allow service provider to know if customer allows for asymmetric bandwidth access like ADSL. It can also be used to rate-limit in a different way upload and download on a symmetric bandwidth access.

The bandwidth is a service bandwidth : expressed primarly as IP bandwidth but if the customer enables MPLS for carrier's carrier, this becomes MPLS bandwidth.

5.11.2. 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 : QoS scheduling profile to be applied.

5.11.2.1. 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 or application and set the appropriate target class of service (target-class-id). The user can define the match using an application reference or a more specific flow definition (based layer 3 source and destination address, layer 4 ports, layer 4 protocol). The current model defines some applications but new application identities may be added through augmentation. The exact meaning of each application identity is up to the service provider, so it will be necessary for the service provider to advise customer on usage of application matching.

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.

```
                                  Provider network
                             +-----------------------+
      192.0.2.0/24
   198.51.100.0/24 ---- CE --------- PE

     Traffic flow
    ---------->
```

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>
            <rule>
                <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>
            </rule>
            <rule>
                <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>
            </rule>
            <rule>
                <id>3</id>
                <match-application>p2p</match-application>
                <target-class-id>DATA3</target-class-id>
            </rule>
            <rule>
                <id>4</id>
                <target-class-id>DATA1</target-class-id>
            </rule>
        </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.

- Peer to peer traffic wille be classified in DATA3.

- 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.

5.11.2.2. QoS profile

User can choose between standard profile provided by the operator or custom profile. The qos-profile defines the traffic scheduling policy to be used by the service provider.

```
                                  Provider network
                             +-----------------------+
   192.0.2.0/24
   198.51.100.0/24 ---- CE --------- PE
                           \       /
                          qos-profile
```

In case of provider managed or co-managed connection, the provider should ensure scheduling according to the requested policy in both traffic directions (SP to customer and customer to SP). As example of implementation, a device scheduling policy may be implemented both at PE and CE side on the WAN link. In case of customer managed connection, the provider is only responsible to ensure scheduling from SP network to the customer site. As example of implementation, a device scheduling policy may be implemented only at PE side on the WAN link towards the customer.

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. When the qos-profile is implemented at CE side the svc-output-bandwidth is taken into account as reference. When it is implemented at PE side, the svc-input-bandwidth is used.

- 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 can be 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. When the qos-profile is implemented at CE side the svc-output-bandwidth is taken into account as reference. When it is implemented at PE side, the svc-input-bandwidth is used.

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.

5.11.3. 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.

5.12. Enhanced VPN features

5.12.1. Carrier's Carrier

In case of Carrier's Carrier ([RFC4364]), a customer MAY want to build MPLS service using an IPVPN as transport layer.

```
           LAN customer1
               |
               |
              CE1
               |
               | -------------
            (vrf_cust1)
             CE1_ISP1
               |                 ISP1 PoP
               | MPLS link
               | -------------
               |
            (vrf ISP1)
              PE1


             (...)               Provider backbone


              PE2
             (vrf ISP1)
               |
               | ------------
               |
               | MPLS link
               |                 ISP1 PoP
              CE2_ISP1
              (vrf_cust1)
               |-------------
               |
              CE2
               |
            Lan customer1
```

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 "carrierscarrier" 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.

In case Carrier's Carrier is enabled, the requested svc-mtu will refer to the MPLS MTU and no more to the IP MTU.

5.12.2. 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 customer 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 provider PoPs. This requires the customer sites to be multihomed. constraint-opaque-value MAY be used to encode additional site location that must be avoided.

5.13. External ID references

The service model sometimes refers to external information through identifiers. As an example, to order a cloud-access to a particular Cloud Service Provider (CSP), the model uses an identifier to refer to the targeted CSP. In case, a customer is using directly this service model as an API (through REST or NETCONF for example) to order a particular service, the service provider should provide a list of authorized identifiers. In case of cloud-access, the service provider will provide the identifiers associated of each available CSP. The same applies to other identifiers like std-qos-profile, oam profile-name, provider-profile for encryption ...

How SP provides those identifiers meaning to the customer is out of scope of this document.

5.14. Defining NNIs

An autonomous system is a single network or group of networks that is controlled by a common system administration group and that uses a single, clearly defined routing protocol. In some cases, VPNs need to span across different autonomous systems in different geographic areas or across different service providers. The connection between autonomous systems is established by the Service Providers and is seamless to the customer.

Some examples are : Partnership between service providers (transport, cloud ...) to extend their VPN service seamlessly, or internal administrative boundary within a single service provider (Backhaul vs Core vs Datacenter ...).

NNIs (Network to Network Interfaces) have to be defined to extend the VPNs across multiple autonomous systems.

[RFC4364] defines multiple flavor of VPN NNI implementations. Each implementation has different pros/cons that are outside the scope of this document. As an example : In an Inter-AS Option A, ASBR peers are connected by multiple interfaces with at least one interface VPN that spans the two autonomous systems. These ASBRs associate each interface with a VPN routing and forwarding (VRF) instance and a Border Gateway Protocol (BGP) session to signal unlabeled IP prefixes. As a result, traffic between the back-to-back VRFs is IP. In this scenario, the VPNs are isolated from each other, and because the traffic is IP, QoS mechanisms that operate on IP traffic can be applied to achieve customer Service Level Agreements (SLAs).

```
     --------                 --------------              -----
    /        \               /              \            /     \
   | Cloud    |             |                |          |       |
   | Provider | ----NNI---- |                | ---NNI---|   DC  |
   |  #1      |             |                |          |       |
    \        /              |                |           \     /
     --------               |                |             ----
                            |                |
     --------               |   My network   |           -----------
    /        \              |                |          /           \
   | Cloud    |             |                |         |             |
   | Provider | ----NNI---- |                |---NNI---|  L3VPN      |
   |  #2      |             |                |         |  Partner    |
    \        /              |                |         |             |
     --------               |                |         |             |
                             \              /          |             |
                              --------------            \           /
                                    |                    ----------
                                    |
                                   NNI
                                    |
                                    |
                            -------------------
                           /                   \
                          |                     |
                          |                     |
                          |                     |
                          |     L3VPN partner   |
                          |                     |
                           \                   /
                             ------------------
```

5.14.1. Defining NNI with option A flavor

```
          AS A                                             AS B
    ---------------------                         --------------------
   /                     \                       /                    \
  |                       |                     |                      |
  |                   ++++++++ InterAS link ++++++++                   |
  |                   +      +_____________ +      +                   |
  |                   +  (VRF1)--(VPN1)----(VRF1)  +                   |
  |                   + ASBR +              + ASBR +                   |
  |                   +  (VRF2)--(VPN2)----(VRF2)  +                   |
  |                   +      +______________+      +                   |
  |                   ++++++++              ++++++++                   |
  |                       |                     |                      |
  |                       |                     |                      |
  |                       |                     |                      |
  |                   ++++++++ InterAS link ++++++++                   |
  |                   +      +_____________ +      +                   |
  |                   +  (VRF1)--(VPN1)----(VRF1)  +                   |
  |                   + ASBR +              + ASBR +                   |
  |                   +  (VRF2)--(VPN2)----(VRF2)  +                   |
  |                   +      +______________+      +                   |
  |                   ++++++++              ++++++++                   |
  |                       |                     |                      |
  |                       |                     |                      |
   \                     /                       \                    /
     --------------------                          -------------------
```

In option A, the two ASes are connected between each other with physical links on Autonomous System Border Routers (ASBR). There may be multiple physical connections between the ASes for a resiliency purpose. A VPN connection, physical or logical (on top of physical), is created for each VPN that needs to cross the AS boundary. A back-to-back VRF model is so created.

This VPN connection can be seen as a site from a service model perspective. Let's say that AS B wants to extend some VPN connection for VPN C on AS A. Administrator of AS B can use this service model to order a site on AS A. All connection scenarios could be realized using the current model features. As an example, the figure above, where two physical connections are involved with logical connections per VPN on top, could be seen as a dual-homed subvpn scenario. And for example, administrator from AS B will be able to choose the appropriate routing protocol (e.g. ebgp) to dynamically exchange routes between ASes.

This document so supposes that option A NNI flavor SHOULD reuse the existing VPN site modeling.

Example : a customer wants from its cloud service provider A to attach its virtual network N to an existing IPVPN (VPN1) he has from a L3VPN service provider B.

```
   Cloud Service provider                  L3VPN SP B
           A
  -----------------                   --------------------
 /                 \                  /                    \
|       |           |                |                      |
|  VM --|       ++++++++  NNI    ++++++++                   |---- VPN1
|       |       +      +_________+      +                   |     Site#1
|       |--------(VRF1)---(VPN1)--(VRF1)+                   |
|       |       + ASBR +         + ASBR +                   |
|       |       +      +_________+      +                   |
|       |       ++++++++         ++++++++                   |
|  VM --|           |                |                      |---- VPN1
|       |Virtual    |                |                      |     Site#2
|       |Network    |                |                      |
|  VM --|           |                |                      |---- VPN1
|       |           |                |                      |     Site#3
 \                 /                  \                    /
   ----------------                    -------------------
                                                 |
                                                 |
                                               VPN1
                                              Site#4
```

The cloud service provider or the customer itself may use our L3VPN service model exposed by service provider B to create the VPN connectivity. We could consider that, as the NNI is shared, the physical connection (bearer) between CSP A and SP B already exists. CSPA may so request through a service model a new site creation with a single site-network-access (single homing used in the diagram). As placement constraint, CSP A may use the existing bearer reference it has from SP A to force the placement of the VPN NNI on the existing link. The XML below describes what could be the configuration request to SP B :

```
<site>
    <site-id>CSP_B_attachment</site-id>
    <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</address-family>
        </bgp>
      </routing-protocol>
    </routing-protocols>
    <site-network-accesses>
     <site-network-access>
      <site-network-access-id>CSP_B_VN1</site-network-access-id>
       <ip-connection>
        <ipv4>
         <address-allocation-type>static-address</address-allocation-type>
         <addresses>
          <provider-address>203.0.113.1</provider-address>
          <customer-address>203.0.113.2</customer-address>
          <mask>30</mask>
         </addresses>
        </ipv4>
       </ip-connection>
       <vpn-attachment>
        <vpn-id>VPN1</vpn-id>
        <site-role>any-to-any-role</site-role>
       </vpn-attachment>
     </site-network-access>
    </site-network-accesses>
    <management>
        <type>customer-managed</type>
    </management>
    <service>
        <svc-input-bandwidth>450000000</svc-input-bandwidth>
        <svc-output-bandwidth>450000000</svc-output-bandwidth>
    </service>
</site>
```

The case described above is different from the cloud-access container usage as the cloud-access provides a public cloud access while this example enables access to private resources located in a cloud service provider network.

5.14.2. Defining NNI with option B flavor

```
          AS A                                             AS B
    ---------------------                         --------------------
   /                     \                       /                    \
  |                       |                     |                      |
  |                   ++++++++ InterAS link ++++++++                   |
  |                   +      +_____________ +      +                   |
  |                   +      +              +      +                   |
  |                   + ASBR +<---mpebgp--->+ ASBR +                   |
  |                   +      +              +      +                   |
  |                   +      +______________+      +                   |
  |                   ++++++++              ++++++++                   |
  |                       |                     |                      |
  |                       |                     |                      |
  |                       |                     |                      |
  |                   ++++++++ InterAS link ++++++++                   |
  |                   +      +_____________ +      +                   |
  |                   +      +              +      +                   |
  |                   + ASBR +<---mpebgp--->+ ASBR +                   |
  |                   +      +              +      +                   |
  |                   +      +______________+      +                   |
  |                   ++++++++              ++++++++                   |
  |                       |                     |                      |
  |                       |                     |                      |
   \                     /                       \                    /
     --------------------                          -------------------
```

In option B, the two ASes are connected between each other with physical links on Autonomous System Border Routers (ASBR). There may be multiple physical connections between the ASes for a resiliency purpose. The VPN "connection" between ASes is done by exchanging VPN routes through MP-BGP.

There are multiple flavors of implementations of such NNI, for example :

1. The NNI is a provider internal NNI between for example of backbone and a DC. There is enough trust between the domains to not filter the VPN routes. So all the VPN routes are exchanged. Route target filtering may be implemented to save some unnecessary route states.

2. The NNI is used between providers that agreed to exchange VPN routes for specific route-targets only. Each provider is authorized to use the route-target values from the other provider.

3. The NNI is used between providers that agreed to exchange VPN routes for specific route-targets only. Each provider has its own route-target scheme. So a customer spanning the two networks will have different route-target in each network for a particular VPN.

Case 1 does not require any service modeling, as the protocol enables dynamic exchange of necessary VPN routes.

Case 2 requires to maintain some route-target filtering policy on ASBRs. From a service modeling point of view, it is necessary to agree on the list of route target to authorize.

In case 3, both ASes need to agree on the VPN route-target to exchange and in addition how to map a VPN route-target from AS A to the corresponding route-target in AS B (and vice-versa).

Those modelings are currently out of scope of this document.

5.14.3. Defining NNI with option C flavor

```
          AS A                                             AS B
    ---------------------                         --------------------
   /                     \                       /                    \
  |                       |                     |                      |
  |                       |                     |                      |
  |                       |                     |                      |
  |                   ++++++++ Multihop ebgp++++++++                   |
  |                   +      +              +      +                   |
  |                   +      +              +      +                   |
  |                   + RGW  +<---mpebgp--->+ RGW  +                   |
  |                   +      +              +      +                   |
  |                   +      +              +      +                   |
  |                   ++++++++              ++++++++                   |
  |                       |                     |                      |
  |                       |                     |                      |
  |                       |                     |                      |
  |                       |                     |                      |
  |                       |                     |                      |
  |                   ++++++++ InterAS link ++++++++                   |
  |                   +      +_____________ +      +                   |
  |                   +      +              +      +                   |
  |                   + ASBR +              + ASBR +                   |
  |                   +      +              +      +                   |
  |                   +      +______________+      +                   |
  |                   ++++++++              ++++++++                   |
  |                       |                     |                      |
  |                       |                     |                      |
  |                       |                     |                      |
  |                   ++++++++ InterAS link ++++++++                   |
  |                   +      +_____________ +      +                   |
  |                   +      +              +      +                   |
  |                   + ASBR +              + ASBR +                   |
  |                   +      +              +      +                   |
  |                   +      +______________+      +                   |
  |                   ++++++++              ++++++++                   |
  |                       |                     |                      |
  |                       |                     |                      |
   \                     /                       \                    /
     --------------------                          -------------------
```

From a VPN service perspective, option C NNI is very similar to option B as a MP-BGP session is used to exchange VPN routes between the ASes. The difference is that the forwarding and control plane are separated on different nodes, so the MP-BGP is multi-hop between routing gateway (RGW) nodes.

Modeling option B and C will be almost identical.

5.15. 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 site-network-access container of a site, only site-network-access 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.

```
<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>
```

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.

```
   +--------------------------------------------------------------+
   |   Hub_Site  ------ PE1               PE2 ------ Spoke_Site1 |
   |      |                   +-----------------------------------+
   |      |                   |
   |      |                   +-----------------------------------+
   |   Hub_Site  ------ PE3               PE4 ------ Spoke_Site2 |
   +--------------------------------------------------------------+
```

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>
    <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</address-family>
            <address-family>ipv6</address-family>
        </bgp>
      </routing-protocol>
    </routing-protocols>
    <site-network-accesses>
        <site-network-access>
            <site-network-access-id>Spoke_Site1</site-network-access-id>
               <access-diversity>
                <groups>
```

```
                 <group>
                  <group-id>20</group-id>
                 </group>
                </groups>
                <constraints>
                 <constraint>
                  <constraint-type>pe-diverse</constraint-type>
                  <target>
                   <group>
                    <group-id>10</group-id>
                   </group>
                  </target>
                 </constraint>
                </constraints>
               </access-diversity>
            <ip-connection>
                <ipv4>
                    <address-allocation-type>static-address</address-
allocation-type>
                    <addresses>
                      <provider-address>203.0.113.254</provider-address>
                      <customer-address>203.0.113.2</customer-address>
                      <mask>24</mask>
                    </addresses>
                </ipv4>
                <ipv6>
                    <address-allocation-type>static-address</address-
allocation-type>
                    <addresses>
                      <provider-address>2001:db8::1</provider-address>
                      <customer-address>2001:db8::2</customer-address>
                      <mask>64</mask>
                    </addresses>
                </ipv6>
            </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>
    <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 5.5, the management system SHOULD use the location information and SHOULD use the access-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 access-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 chosen by the management system. One address will be picked from an allocated subnet for the PE, another will be used for the CE configuration. Routing protocols will also be configured between PE and CE and due to provider managed model, the choice is up to service provider : BGP was chosen for the example. This choice is independant of the routing protocol chosen by customer. For the CE - LAN part, bgp will be used as requested in

the service model. Peering addresses will be derived from those of the connection. 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
 ipv6 address 2001:db8::10:1/64
 ip access-group STD-PROTECT-IN     <---- Standard SP config
!
router bgp 100
 address-family ipv4 vrf Customer1
  neighbor 198.51.100.2 remote-as 65000   <---- Comes from
                                              automated allocation
  neighbor 198.51.100.2 route-map STD in  <---- Standard SP config
  neighbor 198.51.100.2 filter-list 10 in <---- Standard SP config
!
 address-family ipv6 vrf Customer1
  neighbor 2001:db8::0A10:2 remote-as 65000   <---- Comes from
                                              automated allocation
  neighbor 2001:db8::0A10:2 route-map STD in  <---- Standard SP config
  neighbor 2001:db8::0A10:2 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
    ipv6 address 2001:db8::0A10:2/64
   !
   interface FastEthernet11
    description "LAN"
    ip address 203.0.113.254 255.255.255.0 <---- Comes from
                                             ip-connection
    ipv6 address 2001:db8::1/64
   !
   router bgp 65000
    address-family ipv4
     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
    address-family ipv6
     redistribute static route-map STATIC2BGP <---- Standard SP
                                                       configuration
     neighbor 2001:db8::0A10:1 remote-as 100     <---- Comes from
                                                 automated allocation
     neighbor 2001:db8::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.

```
         L3VPN-SVC         |
           service model   |
                           |
                +----------------------+
                |   Service component  | service datastore
                +----------------------+
                           |
                           |
                +----------------------+
           +----|   Config component   |-------+
          /     +----------------------+        \   Network
         /            /            \             \  Configuration
        /            /              \             \ models
       /            /                \             \
+++++++         ++++++++             ++++++++       +++++++
+ CEA + ------- + PE A +             + PE B + ----- + CEB + Config
+++++++         ++++++++             ++++++++       +++++++ datastore

Site A                                       Site B
```

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>
       <addresses>
        <provider-address>203.0.113.254</provider-address>
        <customer-address>203.0.113.2</customer-address>
        <mask>24</mask>
       </addresses>
      </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>203.0.113.254</ip:ip>
        <ip:prefix-length>24</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@2016-06-09.yang"

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

    prefix l3vpn-svc;

    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 2016-06-09 {
        description
         "Traffic protection moved to site level.
          Decouple operational-requirements in two containers.
         ";
        reference "draft-ietf-l3sm-l3vpn-service-yang-08";
    }
    revision 2016-06-06 {
        description
         "Set config false to actual-site-start and stop
          Add a container before cloud-access list
          Add a container before authorized-sites list
          Add a container before denied-sites list
          Modified access-diversity modeling
          Replacing type placement diversity by an identity";
        reference "draft-ietf-l3sm-l3vpn-service-yang-07";
    }
    revision 2016-04-19 {
        description
         "* remove reference to core routing model :
            created new address family identities
          * added features
          * Modified bearer parameters
          * Modified union for ipv4/ipv6 addresses to ip-address
          type
          * Add BSR parameters for multicast
          * Add applications matching for QoS classification
          ";
        reference "draft-ietf-l3sm-l3vpn-service-yang-06";
    }
    revision 2016-04-05 {
        description
         "
         * Added linecard diverse for site diversity
         * Added a new diversity enum in placement-diversity : none
         * Added state to site location

         ";
        reference "";
    }
    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";
    }

    /* Features */

    feature cloud-access {
        description
         "Allow VPN to connect to a Cloud Service
         provider.";
    }
    feature multicast {
        description
        "Enables multicast capabilities in a VPN";
    }
    feature ipv4 {
        description
        "Enables IPv4 support in a VPN";
    }
    feature ipv6 {
        description
        "Enables IPv6 support in a VPN";
    }
    feature carrierscarrier {
        description
        "Enables support of carrier's carrier";
    }
    feature traffic-engineering {
        description
        "Enables support of transport constraint.";
    }
    feature traffic-engineering-multicast {
        description
        "Enables support of transport constraint
        for multicast.";
    }
```

```
    feature extranet-vpn {
        description
        "Enables support of extranet VPNs";
    }
    feature site-diversity {
        description
        "Enables support of site diversity constraints";
    }
    feature encryption {
        description
        "Enables support of encryption";
    }
    feature qos {
        description
        "Enables support of Class of Services";
    }
    feature qos-custom {
        description
        "Enables support of custom qos profile";
    }
    feature rtg-bgp {
        description
        "Enables support of BGP routing protocol.";
    }
    feature rtg-rip {
        description
        "Enables support of RIP routing protocol.";
    }
    feature rtg-ospf {
        description
        "Enables support of OSPF routing protocol.";
    }
    feature rtg-ospf-sham-link {
        description
        "Enables support of OSPF sham-links.";
    }
    feature rtg-vrrp {
        description
        "Enables support of VRRP routing protocol.";
    }
    feature fast-reroute {
        description
        "Enables support of Fast Reroute.";
    }
    feature bfd {
        description
        "Enables support of BFD.";
    }
```

```
    feature always-on {
        description
        "Enables support for always-on access
        constraint.";
    }
    feature requested-type {
        description
        "Enables support for requested-type access
        constraint.";
    }
    feature bearer-reference {
        description
        "Enables support for bearer-reference access
        constraint.";
    }

    /* 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.";
    }



    /* Identities */

    identity placement-diversity {
        description
         "Base identity for site placement
         constraints";
    }
    identity pe-diverse {
        base placement-diversity;
        description
        "Identity for PE diversity";
    }
    identity pop-diverse {
        base placement-diversity;
```

```
        description
        "Identity for POP diversity";
    }
    identity linecard-diverse {
        base placement-diversity;
        description
        "Identity for linecard diversity";
    }
    identity same-pe {
        base placement-diversity;
        description
        "Identity for having sites connected
        on the same PE";
    }
    identity same-bearer {
        base placement-diversity;
        description
        "Identity for having sites connected
        using the same bearer";
    }

    identity customer-application {
        description
         "Base identity for customer application";
    }
    identity web {
        base customer-application;
        description
         "Identity for web application (e.g. HTTP,HTTPS)";
    }
    identity mail {
        base customer-application;
        description
         "Identity for mail applications";
    }
    identity file-transfer {
        base customer-application;
        description
         "Identity for file transfer applications (
         e.g. FTP, SFTP, ...)";
    }
    identity database {
        base customer-application;
        description
         "Identity for database applications";
    }
    identity social {
        base customer-application;
```

```
        description
         "Identity for social network applications";
    }
    identity games {
        base customer-application;
        description
         "Identity for gaming applications";
    }
    identity p2p {
        base customer-application;
        description
         "Identity for peer to peer applications";
    }
    identity network-management {
        base customer-application;
        description
         "Identity for management applications (e.g. telnet
            syslog, snmp ...)";
    }
    identity voice {
        base customer-application;
        description
         "Identity for voice applications";
    }
    identity video {
        base customer-application;
        description
         "Identity for video conference applications";
    }


    identity address-family {
        description
         "Base identity for an address family.";
    }
    identity ipv4 {
        base address-family;
        description
        "Identity for IPv4 address family.";
    }
    identity ipv6 {
        base address-family;
        description
        "Identity for IPv6 address family.";
    }


    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 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 co-managed {
        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-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 {
           container cloud-accesses {
           list cloud-access {
               if-feature cloud-access;
               key cloud-identifier;

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

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

                       leaf site-id {
                           type leafref {
                               path "/l3vpn-svc/sites/site/site-id";
                           }
```

```
                        description
                         "Site ID.";
                    }
                    description
                     "List of denied sites.";
                }
                description
                "Configuration 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
             "Container for cloud access configurations";
        }
        description
         "grouping for vpn cloud definition";
    }

    grouping multicast-rp-group-cfg {
        choice group-format {
            case startend {
                leaf group-start {
                    type inet:ip-address;
                    description
                     "First group address.";
                }
                leaf group-end {
                    type inet:ip-address;
                    description
                     "Last group address.";
                }
            }
            case singleaddress {
                leaf group-address {
```

```
                    type inet:ip-address;
                    description
                     "Group address";
                }
            }
            description
             "Choice for group format.";
        }
        description
         "Definition of groups for
         RP to group mapping.";
    }

    grouping vpn-service-multicast {
        container multicast {
            if-feature 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 {
                container rp-group-mappings {
                    list rp-group-mapping {
                        key "id";

                        leaf id {
                            type uint16;
                            description
                             "Unique identifier for the mapping.";
```

```
                                }
                                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 rp-redundancy {
                                        when "../enabled = true" {
                                            description
                                             "Relevant when RP
                                             is provider managed.";
                                        }
                                        type boolean;
                                        default false;
                                        description
                                         "If true, redundancy
                                         mechanism for RP is required.";
                                    }
                                    leaf optimal-traffic-delivery {
                                        when "../enabled = true" {
                                            description
                                             "Relevant when RP
                                             is provider managed.";
                                        }
                                        type boolean;
                                        default false;
                                        description
                                         "If true, SP must ensure
                                         that traffic uses an optimal path.";
                                    }
                                    description
                                     "Parameters for provider managed RP.";
                                }

                                leaf rp-address {
                                    when "../provider-managed/enabled=false" {
                                        description
                                         "Relevant when RP
                                         is provider managed.";
                                    }
                                    type inet:ip-address;
```

```
                            description
                            "Defines the address of the
                            RendezvousPoint.
                            Used if RP is customer managed.";
                        }

                        container groups {
                            list group {
                                key id;

                                leaf id {
                                    type uint16;
                                    description
                                     "Identifier for the group.";
                                }
                                uses multicast-rp-group-cfg;
                                description
                                "List of groups.";
                            }
                            description
                             "Multicast groups associated with RP.";
                        }

                        description
                         "List of RP to group mappings.";
                    }
                    description
                    "RP to group mappings.";
                }
                container rp-discovery {
                    leaf rp-discovery-type {
                        type identityref {
                            base multicast-rp-discovery-type;
                        }
                        default static-rp;
                        description
                         "Type of RP discovery used.";
                    }
                    container bsr-candidates {
                        when "../rp-discovery-type=bsr-rp" {
                            description
                             "Only applicable if discovery type
                             is BSR-RP";
                        }
                        list bsr-candidate {
                            key address;

                            leaf address {
```

```
                                type inet:ip-address;
                                description
                                 "Address of BSR candidate";
                            }

                            description
                             "List of customer BSR candidates";
                        }
                        description
                         "Customer BSR candidates address";
                    }
                    description
                     "RP discovery parameters";
                }

                description
                 "RendezvousPoint parameters.";
            }
            description
                "Multicast global parameters for the VPN service.";
        }
        description
         "grouping for multicast vpn definition";
    }

    grouping vpn-service-mpls {
        leaf carrierscarrier {
            if-feature carrierscarrier;
            type boolean;
            default false;
            description
             "The VPN is using Carrier's 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 state {
                    type string;
                    description
                    "State of the site.
                    This leaf can also be used
                    to describe a region
                    for country who does not have
                    states.
                    ";
                }
                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 {
                if-feature site-diversity;

                container groups {
                    list group {
                        key group-id;

                        leaf group-id {
                            type string;
                            description
                             "Group-id the site
                             is belonging to";
                        }
                        description
```

```
                        "List of group-id";
                    }
                    description
                     "Groups the site
                     is belonging to.
                     All site network accesses will
                     inherit those group values.";
                }
                description
                     "Diversity constraint type.";
            }
        description
         "This grouping defines site diversity
          parameters";
    }
    grouping access-diversity {
        container access-diversity {
                if-feature site-diversity;
                container groups {
                    list group {
                        key group-id;

                        leaf group-id {
                            type string;
                            description
                             "Group-id the site network access
                             is belonging to";
                        }
                        description
                        "List of group-id";
                    }
                    description
                     "Groups the site network access
                     is belonging to";
                }
                container constraints {
                    list constraint {
                        key constraint-type;

                        leaf constraint-type {
                            type identityref {
                                base placement-diversity;
                            }
                            description
                             "Diversity constraint type.";
                        }
                        container target {
                            choice target-flavor {
```

```
                                case id {
                                    list group {
                                        key group-id;

                                        leaf group-id {
                                            type string;
                                            description
                                             "The constraint will apply
                                             against this particular
                                             group-id";
                                        }
                                        description
                                         "List of groups";
                                    }
                                }
                                case all-accesses {
                                    leaf all-other-accesses {
                                        type empty;
                                        description
                                         "The constraint will apply
                                         against all other site network
                                         access
                                         of this site";
                                    }
                                }
                                case all-groups {
                                    leaf all-other-groups {
                                        type empty;
                                        description
                                         "The constraint will apply
                                         against all other groups the
                                         customer
                                         is managing";
                                    }
                                }
                                description
                                 "Choice for the group definition";
                            }
                            description
                             "The constraint will apply against
                             this list of groups";
                        }
                        description
                         "List of constraints";
                    }
                    description
                     "Constraints for placing this site
                     network access";
```

```
                }

                description
                     "Diversity parameters.";
            }
        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";
           }
        description
         "This grouping defines some operational parameters
          parameters";
    }
    grouping operational-requirements-ops {
           leaf actual-site-start {
               type yang:date-and-time;
               config false;
               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;
               config false;
```

```
              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-protection {
        container traffic-protection {
            if-feature fast-reroute;
            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 carrierscarrier {
            if-feature carrierscarrier;
            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's
             Carrier.";
        }
        description
         "Defines MPLS service parameters for a site.";
    }
    grouping site-service-qos-profile {
        container qos {
            if-feature qos;
            container qos-classification-policy {
                list rule {
                    key id;
                    ordered-by user;

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

                    choice match-type {
                        case match-flow {
                            uses flow-definition;
                        }
                        case match-application {
                            leaf match-application {
                                type identityref {
                                    base customer-application;
                                }
                                description
                                 "Defines the application
                                 to match.";
                            }
                        }
                        description
                         "Choice for classification";
                    }
```

```
                    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 {
                            if-feature qos-custom;
                            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 {
            if-feature 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 {
            container requested-type {
                if-feature requested-type;
                leaf requested-type {
                    type string;
                    description
                     "Type of requested bearer Ethernet, DSL,
                     Wireless ...
                     Operator specific.";
                }
                leaf strict {
                    type boolean;
                    default false;
                    description
                     "define if the requested-type is a preference
                     or a strict requirement.";
                }
```

```
                description
                 "Container for requested type.";
            }
            leaf always-on {
                if-feature always-on;
                type boolean;
                default true;
                description
                "Request for an always on access type.
                This means no Dial access type for
                example.";
            }
            leaf bearer-reference {
                if-feature bearer-reference;
                type string;
                description
                 "This is an internal reference for the
                 service provider.
                 Used ";
            }
            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.";
```

```
                    }
                    if-feature rtg-ospf;
                    leaf-list address-family {
                        type identityref {
                            base 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.";
                    }
                    container sham-links {
                        if-feature rtg-ospf-sham-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
                         "List of Sham links";
                    }
                    description
                     "OSPF specific configuration.";
                }

                container bgp {
```

```
                    when "../type = 'bgp'" {
                        description
                         "Only applies when
                         protocol is BGP.";
                    }
                    if-feature rtg-bgp;
                    leaf autonomous-system {
                        type uint32;
                        description
                         "AS number.";
                    }
                    leaf-list address-family {
                        type identityref {
                            base 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 {
                            if-feature ipv4;
                            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 {
                            if-feature ipv6;
                            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.";
                    }
                    if-feature rtg-rip;
                    leaf-list address-family {
```

```
                        type identityref {
                            base 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.";
                    }
                    if-feature rtg-vrrp;
                    leaf-list address-family {
                        type identityref {
                            base 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 {
                if-feature ipv4;
                leaf address-allocation-type {
                    type identityref {
                        base address-allocation-type;
                    }
                    default "l3vpn-svc:static-address";
                    description
                     "Defines how addresses are allocated.
                     ";
                }

                container addresses {
                    when
                    "../address-allocation-type = 'l3vpn-svc:static-address'" {
                        description
                         "Only applies when
                         protocol allocation type is static";
                    }
                    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
                     "Describes IP addresses used";
                }


                description
                 "IPv4 specific parameters";

            }
            container ipv6 {
                if-feature ipv6;
```

```
                leaf address-allocation-type {
                    type identityref {
                        base address-allocation-type;
                    }
                    default "l3vpn-svc:static-address";
                    description
                     "Defines how addresses are allocated.
                     ";
                }

                container addresses {
                    when
                    "../address-allocation-type = 'l3vpn-svc:static-address'" {
                        description
                         "Only applies when
                         protocol allocation type is static";
                    }
                    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
                     "Describes IP addresses used";
                }

                description
                 "IPv6 specific parameters";

            }
            container oam {
                container bfd {
                    if-feature 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 {
            if-feature 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 {
                    if-feature ipv4;
                    type boolean;
                    default true;
                    description
                        "Enables ipv4 multicast transport";
                }
                leaf ipv6 {
                    if-feature 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 address-family;
                }
                description
                 "Transport protocol used for management.";
            }
            leaf address {
                type inet:ip-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 {
                                        if-feature ipv4;
                                        key lan;

                                        leaf lan {
                                            type inet:ipv4-prefix;
                                            description
                                             "Lan prefixes.";
                                        }
                                        description "
                                            List of LAN prefixes
                                            for the site.
                                            ";
                                    }
                                    list ipv6-lan-prefixes {
                                        if-feature ipv6;
                                        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 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 {
            if-feature traffic-engineering;
            container unicast-transport-constraints {
                list constraint {
                    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 {
                if-feature traffic-engineering-multicast;
                list constraint {
                    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 {
            if-feature extranet-vpn;
            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 {
            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-protection;
        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;
             uses operational-requirements-ops;

             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

The YANG modules defined in this document MAY be accessed via the RESTCONF protocol [I-D.ietf-netconf-restconf] or NETCONF protocol ([RFC6241]. The lowest RESTCONF or NETCONF layer requires that the transport-layer protocol provides both data integrity and confidentiality, see Section 2 in [I-D.ietf-netconf-restconf] and [RFC6241]. The client MUST carefully examine the certificate presented by the server to determine if it meets the client's expectations, and the server MUST authenticate client access to any protected resource. The client identity derived from the authentication mechanism used is subject to the NETCONF Access Control Module (NACM) ([RFC6536]). Other protocols to access this YANG module are also required to support the similar mechanism.

The data nodes defined in the "ietf-l3vpn-svc" YANG module MUST be carefully created/read/updated/deleted. The entries in the lists below include customer proprietary or confidential information, therefore only authorized clients MUST access the information and the other clients MUST NOT be able to access to the information.

- /l3vpn-svc/vpn-services/vpn-svc

- /l3vpn-svc/sites/site

- /l3vpn-svc/site-templates/site-template

10. Acknowledgements

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

11. IANA Considerations

The IANA is requested to assign a new URI from the IETF XML registry ([RFC3688]). Authors are suggesting the following URI :

```
URI: urn:ietf:params:xml:ns:yang:ietf-l3vpn-svc
Registrant Contact: L3SM WG
XML: N/A, the requested URI is an XML namespace
```

This document also requests a new YANG module name in the YANG Module Names registry ([RFC6020]) with the following suggestion :

```
name: ietf-l3vpn-svc
namespace: urn:ietf:params:xml:ns:yang:ietf-l3vpn-svc
prefix: l3vpn-svc
reference: RFC XXXX
```

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>.

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

[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

[I-D.ietf-netconf-restconf] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", draft-ietf-netconf-restconf-13 (work in progress), April 2016.

[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>.

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <http://www.rfc-editor.org/info/rfc6536>.

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 Services

Email: stephane.litkowski@orange.com

Rob Shakir
Jive Communications

Email: rjs@rob.sh

Luis Tomotaki
Verizon

Email: luis.tomotaki@verizon.com

Kenichi Ogaki
KDDI

Email: ke-oogaki@kddi.com

Kevin D'Souza
ATT

Email: kd6913@att.com